

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

INFAL51

Initiation à Linux

Kevin Hagner

11-13 mars 2020

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

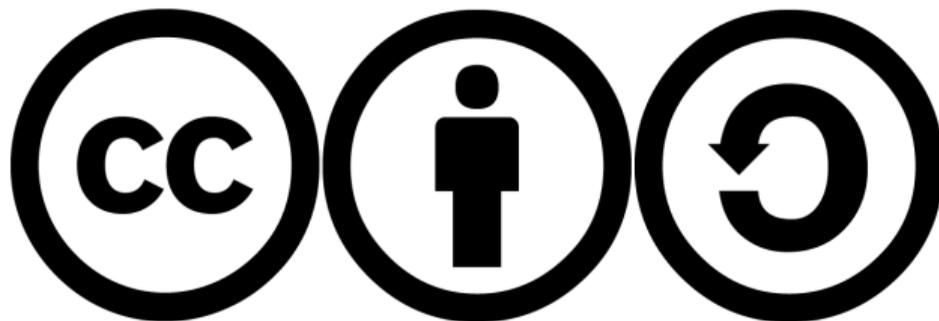
Les sauvegardes

Conclusion

- Développeur Web back-end.
- Administrateur de systèmes Linux.
- Utilisateur enthousiaste de Linux depuis plus de dix ans.
- Cinquième intervention en tant que formateur.

Licence de cette présentation

Figure – CC BY SA



Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs
- Les sauvegardes

Conclusion

Licence de cette présentation

Figure – CC BY SA



- Licence *Creative Commons*[↗].
- Réutilisation libre et gratuite.
- Possibilité d'y faire des modifications.
- Il faudra par contre citer mon prénom et mon nom (clause BY) sur votre création.
- Votre œuvre devra être partagée avec les mêmes conditions (clause SA).

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Contrôle continu

- Récupérez le document : <https://share.spyzone.fr/cesi/infal51.odt>
- Effectuez les exercices en fonction de l'évolution du cours.
- Vous m'enverrez votre questionnaire complété à la fin des trois jours.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Contrôle continu

- Récupérez le document : <https://share.spyzone.fr/cesi/infal51.odt>
- Effectuez les exercices en fonction de l'évolution du cours.
- Vous m'enverrez votre questionnaire complété à la fin des trois jours.

Cours inversés

- Introduction d'un sujet.
- Puis un travail pratique ou de recherche pour l'approfondissement.
- Une conclusion permettant de résumer et consolider les notions évoquées.

Une introduction à Linux

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

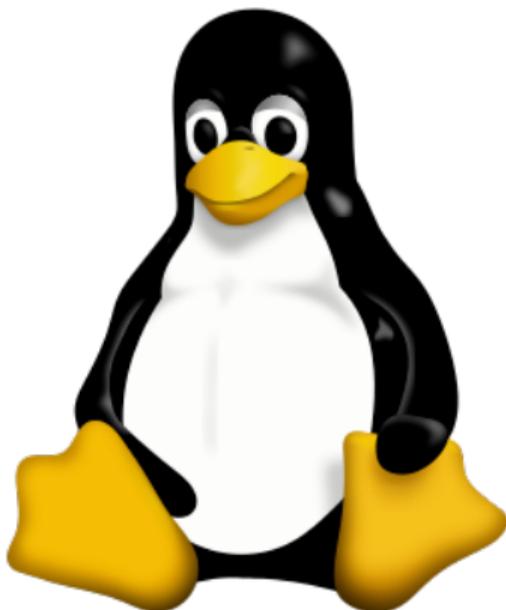
Le stockage

Btrfs

Les sauvegardes

Conclusion

Figure – Tux, la mascotte



Introduction

Contexte

Le noyau
UNIX
Linux
Les distributions
Conclusion

Installation

Xubuntu
Les mises à jour
Gestion des logiciels

Les bases

GNU
Le shell
Les utilisateurs
Les fichiers
Les processus
Les scripts

Les données

Le stockage
Btrfs
Les sauvegardes

Conclusion

1 Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

2 Installation

Xubuntu

Les mises à jour

Gestion des logiciels

3 Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

4 Les données

Le stockage

Btrfs

Les sauvegardes

5 Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 1.1

À quoi sert un noyau ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

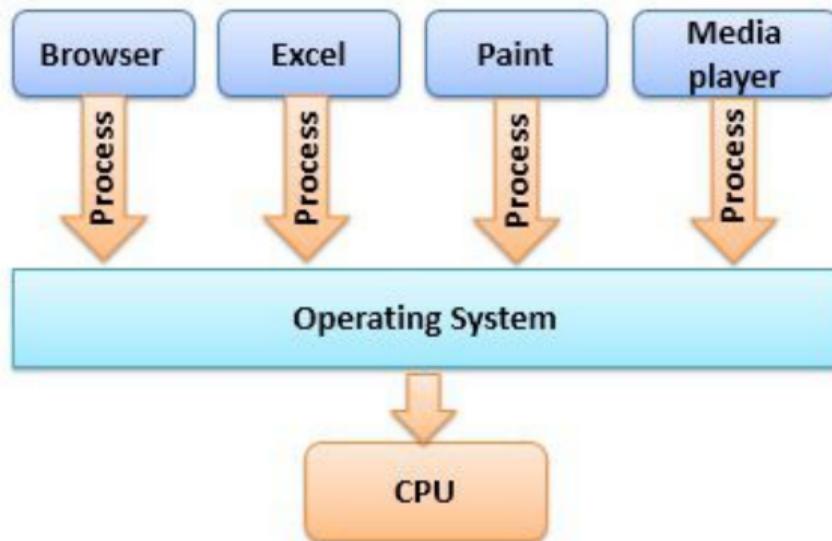
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Figure – Un système multitâches (*crédit*[↗])**Multitasking**

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

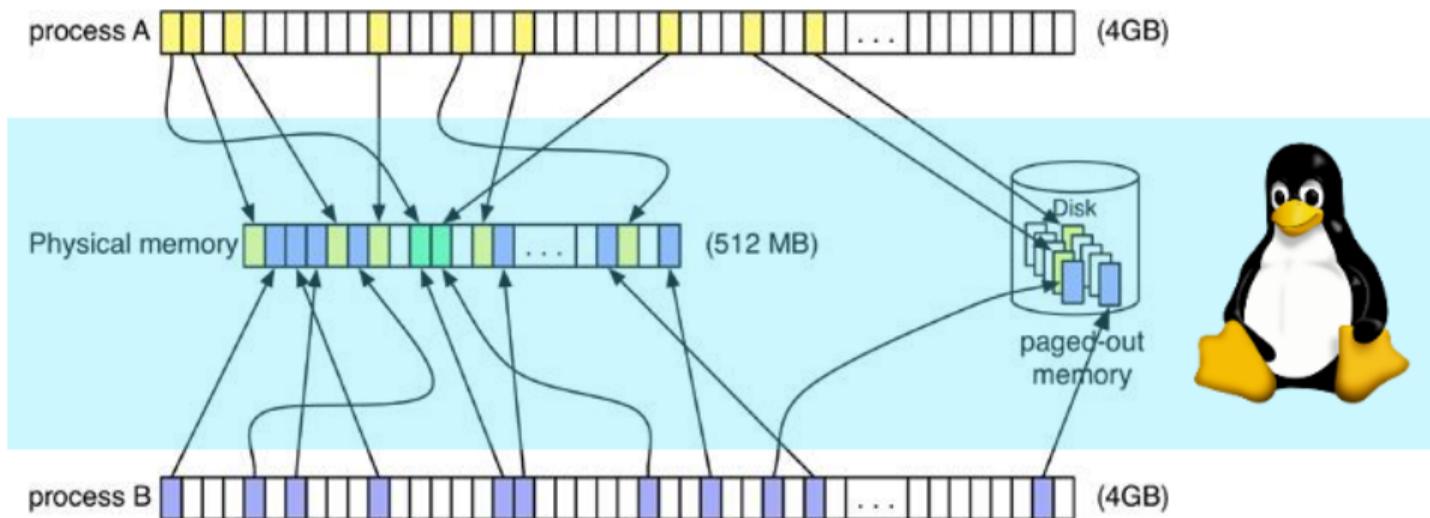
Btrfs

Les sauvegardes

Conclusion

Gestion de la mémoire

Gestion de la mémoire

Figure – Partage de la mémoire physique entre les applications (*crédit*[↗])

Gestion des périphériques

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

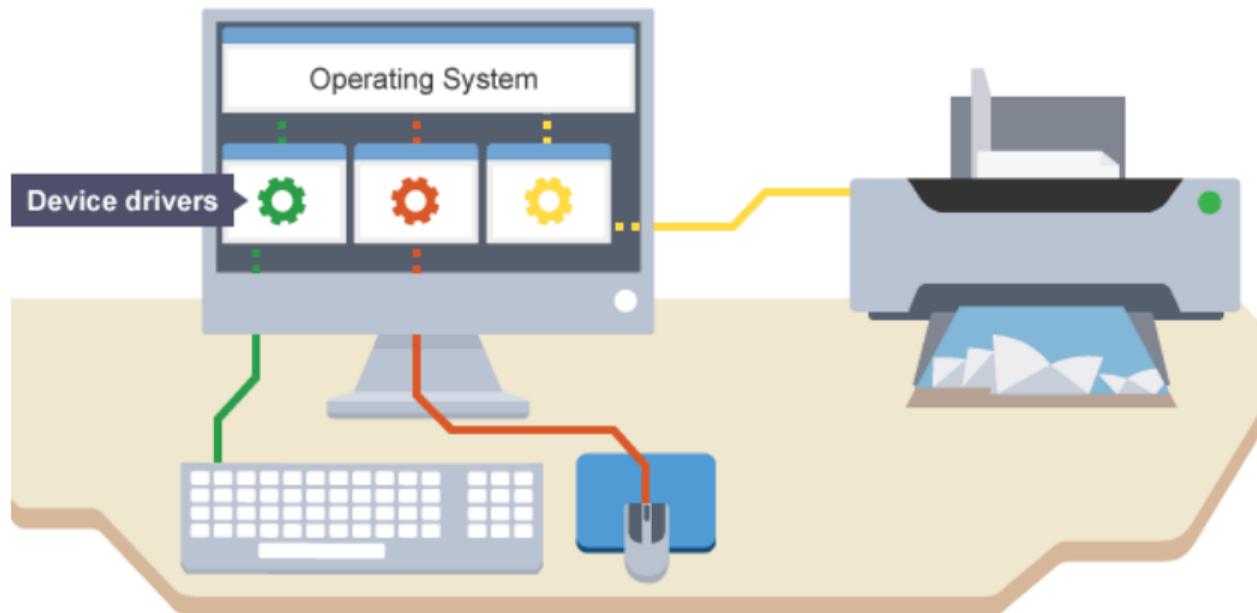
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Figure – Gestion des périphériques à l'aide des drivers (*crédit*[Ⓔ])

API d'utilisation par les processus

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

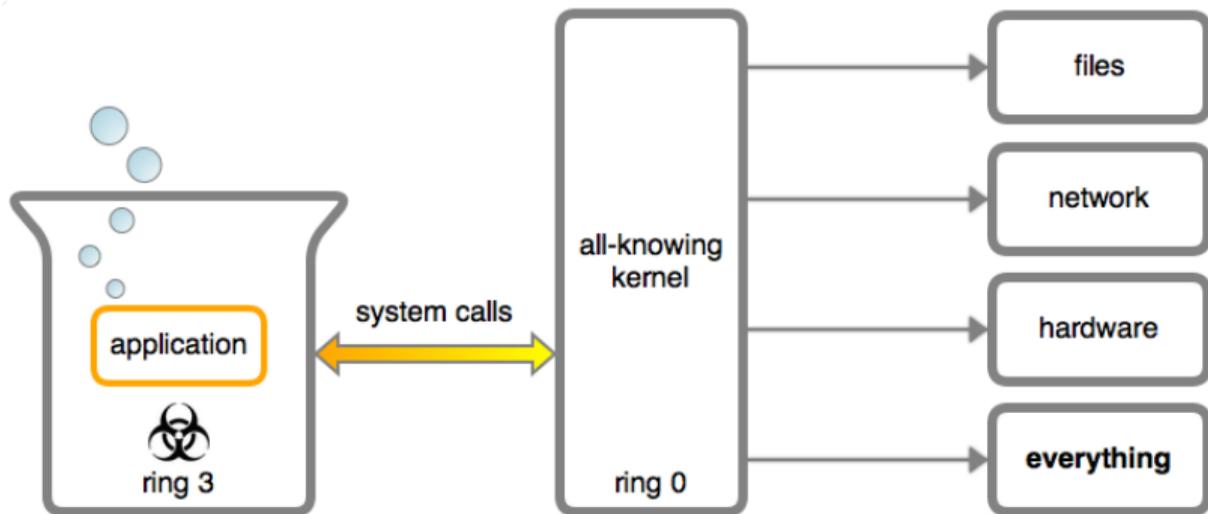
Le stockage

Btrfs

Les sauvegardes

Conclusion

API d'utilisation par les processus

Figure – Le kernel est au service des applications utilisateurs via les *syscalls* (crédit¹)

<http://duartes.org/gustavo/blog>

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 1.2

Le monde d'UNIX

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Ken Thompson



- Né le 4 février 1943 à La Nouvelle-Orléans.
- Il rejoint les Laboratoires Bell en 1966 (à 23 ans).
- Il y travaille sur *Multics*, puis *UNIX* à partir de 1969.
- Invente le langage *B*.
- Participe à la création du *GO*.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Dennis Ritchie



- Né le 9 septembre 1941 à Bronxville.
- Il rejoint les Laboratoires Bell en 1967 (à 26 ans).
- Invente le langage C.
- Mort le 12 octobre 2011.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

1964 Début du projet par le *MIT*, *Bell Labs* et *General Electric*.

1969 *Bell Labs* se retire.

1980 Annonce de la fin du développement.

1990 Le prix Turing est donné à *Fernando Corbató* pour son rôle de leader dans le projet.

2000 Arrêt du dernier système Multics connu.

2006 Le code source devient public.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- 1969 Quand *Bell Labs* abandonna *Multics*, ses développeurs (*Ken Thompson* et *Dennis Ritchie* notamment) décidaient de tenter une réécriture du projet qui se voulait moins complexe.
- 1972 UNIX a été réécrit de l'assembleur au C.
- 1975 Suite à une politique antitrust, *AT&T* dont dépendent les laboratoires *Bell* n'ont pas le droit de commercialiser autre chose que des équipements téléphoniques. L'entreprise fut donc le choix de partager le système et ses sources aux universités à des fins éducatives.
- 1977 *UNIX* commence à se démocratiser dans les milieux universitaires. La *Berkeley Software Distribution* crée sa propre version améliorée du noyau.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- À partir de 1977, une multitude de forks d'UNIX voient le jour un peu partout.
- Normes définies depuis 1988 par l'*IEEE* pour standardiser le fonctionnement d'un système UNIX.
- L'accès au détail et la certification d'un produit sont très chers.
- *Single UNIX Specification*[↗] est une famille de norme qui reprend et complète *POSIX* et qui en plus est disponible gratuitement.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 1.3

Linux

Linus Torvalds

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Linus Torvalds



- Né le 28 décembre 1969 à Helsinki.
- Créateur de *Linux* et de *Git*.
- Étudie à l'université d'Helsinki de 1988 à 1996.
- Travaille actuellement à la Fondation Linux.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Clone d'Unix conçu par *Andrew Tanenbaum* en 1987 comme support pédagogique pour ses étudiants.
- A servi d'inspiration à *Linus* pour la création de *Linux*.
- Toujours développé aujourd'hui.
- Utilisé par l'*Intel Management Engine* (Intel ME). Si vous avez un processeur *Intel*, vous avez donc probablement un *Minix* en fonctionnement actuellement.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La naissance de Linux

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- En 1991, *Linus* constate des faiblesses sur l'émulateur de terminal de *Minix*. Il décide de concevoir le sien, sur une disquette.
- Il devient de plus en plus complexe, jusqu'à supporter *Bash* et *gcc*.
- Le 25 août 1991, *Linus* communique sur sa création sur le newsgroup *comp.os.minix* et qualifie son noyau comme n'étant rien de professionnel et de gros comme le sera *GNU*.
- *Tanenbaum* considérait *Linux* comme obsolète car monolithique face à son microkernel.
- Progressivement, *Linux* prend la tournure d'un logiciel libre et des centaines de développeurs se mettent à contribuer.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Popularité de Linux

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Aujourd'hui, *Linux* est employé en masse dans tous les secteurs touchants à l'informatique, sauf les *PC* grand public. Quelques exemples :

- *Android* utilise *Linux* et est présent dans 74 % des téléphones.
- Une grosse majorité des fournisseurs de services Web (*Google*, *Facebook*, *Twitter*, *Amazon*, etc.).
- Une grosse majorité des fabricants de matériel (*AMD*, *Intel*, *IBM*).
- La plupart des systèmes embarqués.

Les contributions pour la version 4.13

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les contributions pour la version 4.13

- **1681** développeurs différents.
- Représentés par **225** entreprises.
- Plus de **85 %** des contributeurs sont payés par des sociétés tierces pour leur travail sur le noyau.

Entreprises les plus représentées

Pos.	Nom	% chang.	Pos.	Nom	% chang.
1	Intel	13,1%	11	AMD	2,7%
2	<i>Aucune</i>	8,2%	14	Oracle	1,7%
3	Red Hat	7,2%	17	ARM	1,5%
6	IBM	4,1%	21	Canonical	1,0%
8	Samsung	3,2%	22	Facebook	0,9%
10	Google	3,0%	28	NVidia	0,7%

Sources : *Kernel Report 2017*[↗]

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Comment contribuer à Linux ?

Comment contribuer à Linux ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Écrire un patch corrigeant ou apportant une nouvelle fonctionnalité.
- Il est ensuite posté sur une liste de diffusion dans laquelle il sera commenté par les autres développeurs.
- Il doit finir validé par le mainteneur du sous-système en relation avec la modification.
- Finalement, le patch est intégré par *Linus Torvalds*.

Sources : kernel.org[↗]

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Liens utiles

- Lien de téléchargement : kernel.org[↗]
- Liste de diffusion : lkml.org[↗]
- Code source : git.kernel.org[↗]

Exercice 1.4

Les distributions Linux

Comment installer un logiciel à la main ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Comment installer un logiciel à la main ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Téléchargement du code source.
- Configuration d'un environnement permettant sa compilation.
- Compilation du logiciel, et édition d'éventuels patchs si celle-ci échoue.
- Récupération et compilation des dépendances demandées (par exemple, la bibliothèque qui s'occupe d'afficher une fenêtre à l'écran).
- Définir avec précision où déposer ces fichiers compilés de façon à ce qu'ils soient repérés par les autres applications qui en dépendent.
- Écrire les fichiers de configuration et les placer aux bons endroits.
- Utiliser le logiciel... et ne pas oublier de refaire tout ça de temps en temps pour appliquer les mises à jour !

Le rôle d'une distribution

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Faciliter l'interopérabilité entre les applications existantes.

- Système uniforme de management des dépendances.
- Ensemble cohérent et symbiotique de logiciels.
- Gestion simplifiée d'installation et de mises à jour.
- Communautés d'utilisateurs.

Aujourd'hui, il existe *plus d'une centaine*[↗] de distributions différentes.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Red Hat Enterprise



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Red Hat Enterprise



- Distribution commerciale vendue par *Red Hat*.
- Permet d'avoir un support fiable et une assistance technique.
- Largement utilisé sur les serveurs.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



- Distribution commerciale supervisée par *Red Hat*.
- Distribution grand public dont l'installation et l'utilisation se veulent les plus simples possible.
- Le cycle de développement et cours (une nouvelle version majeure tous les six mois environ), les logiciels sont fréquemment mis à jour.
- Employée principalement sur des ordinateurs de bureau.



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



- Distribution non commerciale maintenue par une organisation communautaire et démocratique.
- Debian n'inclut que des logiciels libres. Ceux possédant du code propriétaire ne sont pas acceptés.
- Le cycle de développement est lent, les logiciels ne sont mis à jour de façon régulière que pour répondre à des corrections de bogues ou de faille de stabilité.
- Distribution reconnue pour sa fiabilité, elle sert de base à de nombreux autres (*Knoppix, Ubuntu, SteamOS, etc.*).
- Utilisée autant sur des serveurs que sur des ordinateurs de bureau.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



- Très légère (environ 5 Mo).
- Utilisée en embarqué et pour la création de conteneurs Linux.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



- Distribution basée sur la sécurité, la vie privée et l'anonymat.
- Fonctionne en système *live*, et ne laisse (normalement) pas de trace sur l'ordinateur.

Comment faire son choix ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Des centaines de distributions existent... 

Comment faire son choix ?

Comment faire son choix ?

Des centaines de distributions existent... 

Comment faire son choix ?

- En fonction de la philosophie.
- En fonction de sa popularité.
- Les modes de financement.
- L'utilisation visée.
- Le support logiciel.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Le noyau gère la complexité du matériel.
- Il gère aussi l'interaction entre les logiciels.
- Il présente des interfaces uniformes aux utilisateurs.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Au début était *Multics*, un noyau complexe et difficilement maintenable.
- Puis est venu *UNIX*, une réécriture totale de *Multics*, plus simple.
- *UNIX* a été massivement employé dans le milieu universitaire. *Minix* est l'une de ses (nombreuses !) réimplémentations.
- *Linux* est créé sur le modèle de *Minix*. Son code étant libre, beaucoup d'autres développeurs ont fini par rejoindre le projet.
- Le noyau est finalement utilisé pour intégrer une structure encore plus grosse ayant la prétention de former des systèmes d'exploitation complets : les distributions.

Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu**
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs
- Les sauvegardes

Conclusion

Exercice 2.1

Installation de Xubuntu

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Installation de VirtualBox.
- Création d'une nouvelle machine virtuelle.
 - Allouer au moins 1000 Mio de mémoire vive.
 - Création d'un disque dur d'au moins 12 Gio.
 - Configuration réseau par pont.
- Téléchargement de *Xubuntu 18.04 LTS*[☞].
- Booter la machine virtuelle sur le live CD.
- Installer Xubuntu.
- Faire un clone de l'image pour pouvoir la restaurer en cas de problème.

La virtualisation

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Simulation d'un système d'exploitation dans un autre.
- Pratique pour tester des logiciels sans affecter votre véritable *OS*.
- Pratique aussi pour isoler différentes applications et améliorer ainsi la sécurité.
- Par contre, l'utilisation des ressources physique peut s'avérer moins bonne.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 2.2

Les mises à jour

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

À quoi servent les mises à jour ?

À quoi servent les mises à jour ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

À récupérer les dernières évolutions d'un logiciel

MàJ de sécurité

À quoi servent les mises à jour ?

À récupérer les dernières évolutions d'un logiciel

MàJ de sécurité Corrige une faille compromettant les données de l'utilisateur *CVE*^[4].

MàJ correctives

À quoi servent les mises à jour ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

À récupérer les dernières évolutions d'un logiciel

MàJ de sécurité Corrige une faille compromettant les données de l'utilisateur *CVE*^[1].

MàJ correctives Corrige un bogue qui empêchait une fonctionnalité déjà présente de fonctionner normalement.

MàJ évolutives

À quoi servent les mises à jour ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

À récupérer les dernières évolutions d'un logiciel

MàJ de sécurité Corrige une faille compromettant les données de l'utilisateur *CVE*^[1].

MàJ correctives Corrige un bogue qui empêchait une fonctionnalité déjà présente de fonctionner normalement.

MàJ évolutives Ajoutent une nouvelle fonctionnalité dans un logiciel déjà existant.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les différentes versions de Xubuntu

Les différentes versions de Xubuntu

Version	Date sortie	Fin support	Durée support
16.04 LTS	21/04/2016	21/04/2019	3 ans
...
17.10	19/10/2017	07/2018	1 an
18.04 LTS	26/04/2018	29/04/2021	3 ans
18.10	18/10/2018	18/07/2019	1 an
19.04	18/04/2019	18/01/2020	1 an
19.10	17/10/2019	17/07/2021	1 an

Source : [Wikipedia](#)[🔗]

À quoi sert le support ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jourGestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les différentes versions de Xubuntu

Version	Date sortie	Fin support	Durée support
16.04 LTS	21/04/2016	21/04/2019	3 ans
...
17.10	19/10/2017	07/2018	1 an
18.04 LTS	26/04/2018	29/04/2021	3 ans
18.10	18/10/2018	18/07/2019	1 an
19.04	18/04/2019	18/01/2020	1 an
19.10	17/10/2019	17/07/2021	1 an

Source : [Wikipedia](#)[🔗]

À quoi sert le support ?

À continuer de fournir des mises à jour de sécurité et correctives après la sortie du système d'exploitation. Les mises à jour évolutives y sont par contre exclues.

Versions courtes VS *LTS* VS *Rolling releases*

Versions courtes



Versions à long terme (*LTS*)



Versions courtes VS *LTS* VS *Rolling releases*

Versions courtes

- ⊕ Compromis souvent acceptable entre fiabilité et intégration des nouvelles fonctionnalités.
- ⊖ Nécessite des montées de versions plus souvent, et qui peuvent prendre du temps à l'administrateur.

Versions à long terme (*LTS*)

- ⊕
- ⊕
- ⊖
- ⊖

Versions courtes VS *LTS* VS *Rolling releases*

Versions courtes

- \oplus Compromis souvent acceptable entre fiabilité et intégration des nouvelles fonctionnalités.
- \ominus Nécessite des montées de versions plus souvent, et qui peuvent prendre du temps à l'administrateur.

Versions à long terme (*LTS*)

- \oplus Système généralement très fiable.
- \oplus La maintenance est faible, les *MàJ* ont peu de risques de casser quelque chose.
- \ominus Il faut attendre longtemps avant d'avoir accès aux nouvelles fonctionnalités.
- \ominus Les sauts entre les versions étant gros, les risques lors des transitions sont forts.

Versions courtes VS *LTS* VS *Rolling releases*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Versions roulantes (*Rolling release*)



Versions courtes VS *LTS* VS *Rolling releases*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

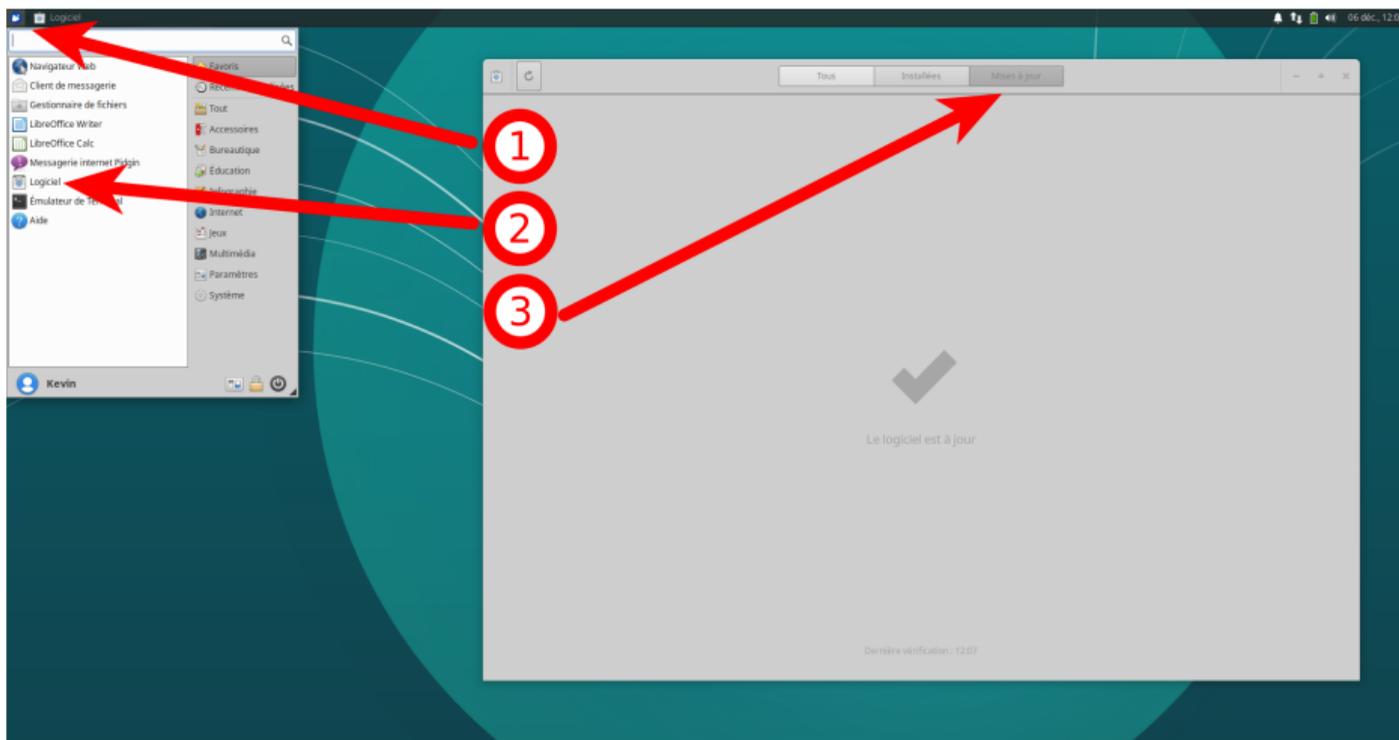
Conclusion

Versions roulantes (*Rolling release*)

- \oplus Aucune montée de version nécessaire : les *MàJ* sont « douces » : les évolutions majeures sont intégrées au rythme des développeurs, pas de la distribution. Il n'y a donc pas de moments T ou tous les logiciels évoluent en même temps.
- \oplus Récupération très rapide des nouvelles fonctionnalités.
- \ominus ... et donc des bogues associés. Ces systèmes sont donc souvent moins fiables et grand public.

Exercice 2.3

Utilisation du gestionnaire de paquets

Mise à jour du système depuis *Logiciel*

Mise à jour du système depuis *apt*

The screenshot shows a Linux desktop environment with a terminal window open. The terminal displays the following commands and output:

```

kevin@kevin-VirtualBox:~$ sudo su
root@kevin-VirtualBox:/home/kevin# apt update && apt upgrade
Atteint :1 http://fr.archive.ubuntu.com/ubuntu bionic InRelease
Atteint :2 http://fr.archive.ubuntu.com/ubuntu bionic-updates InRelease
Atteint :3 http://fr.archive.ubuntu.com/ubuntu bionic-backports InRelease
Atteint :4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
1 paquet peut être mis à jour. Exécutez « apt list --upgradable » pour le voir.
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants seront mis à jour :
  unattended-upgrades
1 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 41,7 ko dans les archives.
Après cette opération, 24,6 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] 

```

Red circles and arrows indicate the following steps:

- Opening the terminal window.
- Running the command `sudo su`.
- Running the command `apt update && apt upgrade`.
- The terminal output showing the progress of the update process.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Installation de *Kapman* depuis *Logiciel*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

**Gestion des
logiciels**

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

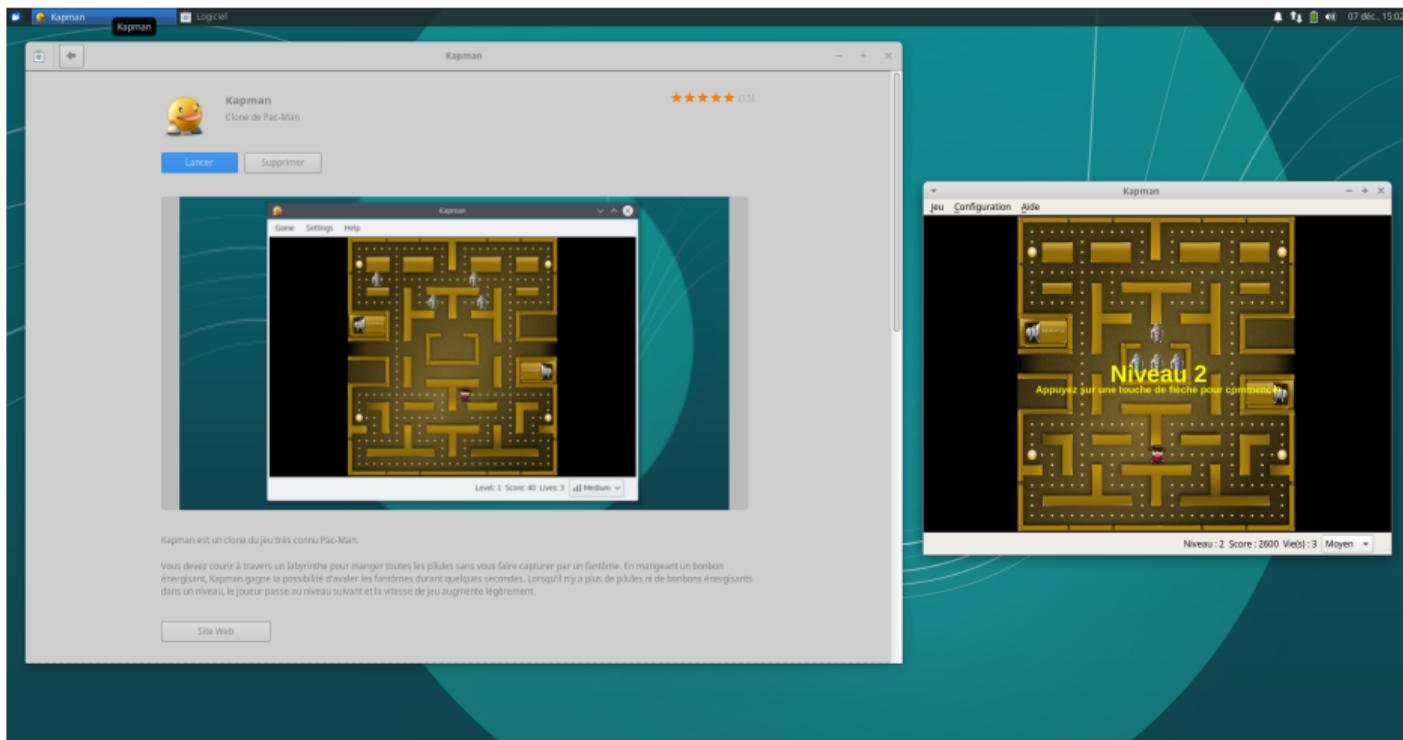
Les données

Le stockage

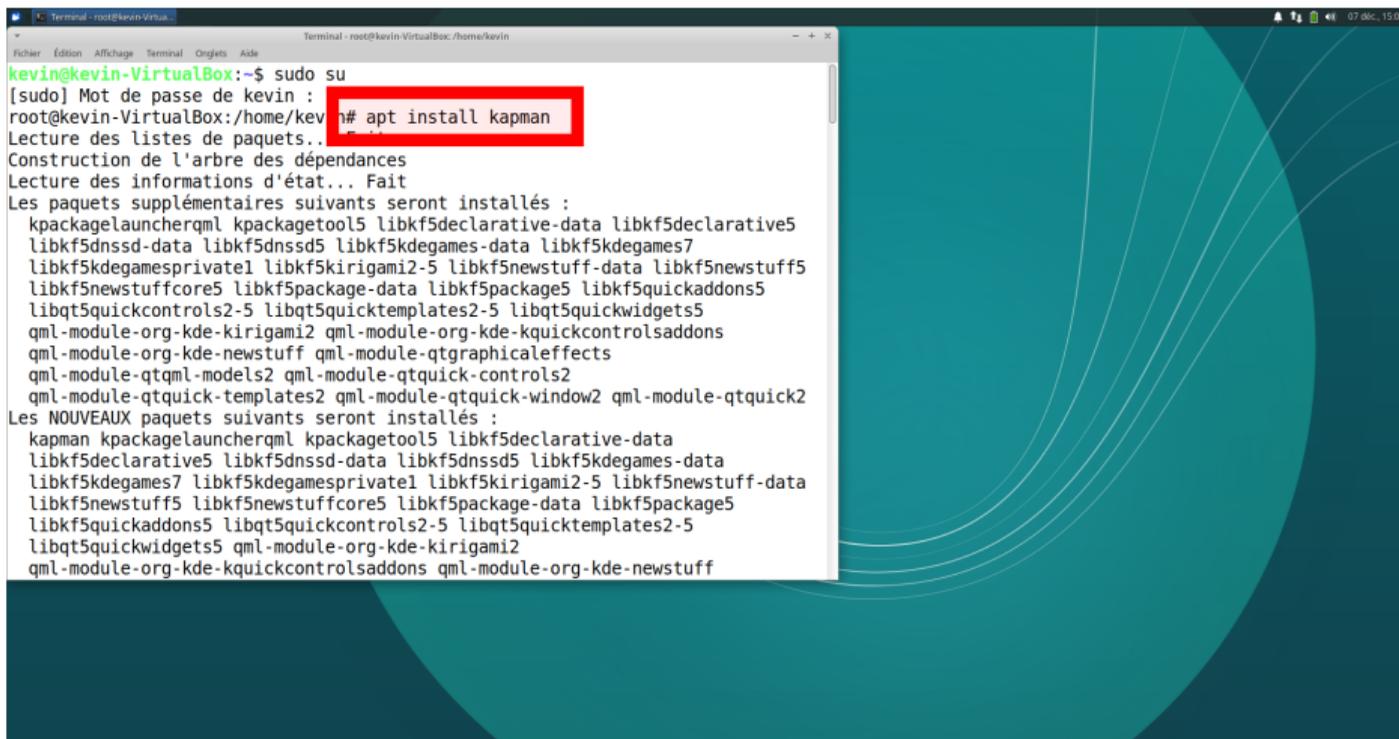
Btrfs

Les sauvegardes

Conclusion



Installation de *Kapman* depuis *apt*



```
Terminal - root@Kevin-VirtualBox
Fichier  Edition  Affichage  Terminal  Onglets  Aide
kevin@kevin-VirtualBox:~$ sudo su
[sudo] Mot de passe de kevin :
root@kevin-VirtualBox:/home/kevin# apt install kapman
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
kpackagelauncherqml kpackagetool5 libkf5declarative-data libkf5declarative5
libkf5dnssd-data libkf5dnssd5 libkf5kdegames-data libkf5kdegames7
libkf5kdegamesprivate1 libkf5kirigami2-5 libkf5newstuff-data libkf5newstuff5
libkf5newstuffcore5 libkf5package-data libkf5package5 libkf5quickaddons5
libqt5quickcontrols2-5 libqt5quicktemplates2-5 libqt5quickwidgets5
qml-module-org-kde-kirigami2 qml-module-org-kde-kquickcontrolsaddons
qml-module-org-kde-newstuff qml-module-qtgraphicaleffects
qml-module-qtqml-models2 qml-module-qtquick-controls2
qml-module-qtquick-templates2 qml-module-qtquick-window2 qml-module-qtquick2
Les NOUVEAUX paquets suivants seront installés :
kapman kpackagelauncherqml kpackagetool5 libkf5declarative-data
libkf5declarative5 libkf5dnssd-data libkf5dnssd5 libkf5kdegames-data
libkf5kdegames7 libkf5kdegamesprivate1 libkf5kirigami2-5 libkf5newstuff-data
libkf5newstuff5 libkf5newstuffcore5 libkf5package-data libkf5package5
libkf5quickaddons5 libqt5quickcontrols2-5 libqt5quicktemplates2-5
libqt5quickwidgets5 qml-module-org-kde-kirigami2
qml-module-org-kde-kquickcontrolsaddons qml-module-org-kde-newstuff
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Où trouver de l'aide pour l'utilisation de *Xubuntu*?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

**Gestion des
logiciels**

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Où trouver de l'aide pour l'utilisation de *Xubuntu*?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- *Le wiki officiel Ubuntu* [↗](#).
- Les sites de la communauté :
 - *Forum Xubuntu* [↗](#)
 - *Forum Ubuntu* [↗](#)
 - *Des salons IRC* [↗](#)
- Les ressources prévues pour d'autres distributions :
 - *Le wiki d'Archlinux* [↗](#)

Les actions d'un gestionnaire de paquets

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

```

root@Kevin-VirtualBox:/home/kevin# apt install vim
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  vim-runtime
Paquets suggérés :
  ctags vim-doc vim-scripts
Les MOUNGAIK paquets suivants seront installés :
  vim vim-runtime
0 mis à jour, 2 nouvellement installés, 0 à enlever et 1 non mis à jour.
Il est nécessaire de prendre 0 507 ko dans les archives.
Après cette opération, 32,0 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] o
Réception de :1 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim-runtime all 2:8.0.1453-1ubuntu1 [5 435 kB]
Réception de :2 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim amd64 2:8.0.1453-1ubuntu1 [1 152 kB]
6 507 ko réceptionnés en 5s (1 437 ko/s)
Sélection du paquet vim-runtime précédemment désélectionné.
(Lecture de la base de données... 191012 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../vim-runtime_2:8.0.1453-1ubuntu1_all.deb ...
Ajout de « détournement de /usr/share/vim/vim80/doc/help.txt en /usr/share/vim/vim80/doc/help.txt.vim-tiny par vim-runtime »
Ajout de « détournement de /usr/share/vim/vim80/doc/tags en /usr/share/vim/vim80/doc/tags.vim-tiny par vim-runtime »
Dépaquetage de vim-runtime (2:8.0.1453-1ubuntu1) ...
Sélection du paquet vim précédemment désélectionné.
Préparation du dépaquetage de .../vim_2:8.0.1453-1ubuntu1_amd64.deb ...
Dépaquetage de vim (2:8.0.1453-1ubuntu1) ...
Paramétrage de vim-runtime (2:8.0.1453-1ubuntu1) ...
Paramétrage de vim (2:8.0.1453-1ubuntu1) ...
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vim » (vim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vimdiff » (vimdiff) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rvim » (rvim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rview » (rview) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vi » (vi) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/view » (view) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/ex » (ex) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.8.3-2ubuntu0.1) ...

```

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les actions d'un gestionnaire de paquets

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

```

root@Kevin-VirtualBox:~/home/kevin# apt install vim
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  vim-runtime
Paquets suggérés :
  ctags vim-doc vim-scripts
Les MOUVABLES paquets suivants seront installés :
  vim vim-runtime
0 mis à jour, 2 nouvellement installés, 0 à enlever et 1 non mis à jour.
Il est nécessaire de prendre 6 587 ko dans les archives.
Après cette opération, 32,0 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] o
Réception de :1 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim-runtime all 2:8.0.1453-1ubuntu1.1 [5 435 kB]
Réception de :2 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim amd64 2:8.0.1453-1ubuntu1.1 [1 152 kB]
6 587 ko réceptionnés en 3s (1 437 ko/s)
Sélection du paquet vim-runtime précédemment désélectionné.
(Lecture de la base de données... 191012 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../vim-runtime_2:8.0.1453-1ubuntu1.1_all.deb ...
Ajout de « détournement de /usr/share/vim/vim80/doc/help.txt en /usr/share/vim/vim80/doc/help.txt.vim-tiny par vim-runtime »
Ajout de « détournement de /usr/share/vim/vim80/doc/tags en /usr/share/vim/vim80/doc/tags.vim-tiny par vim-runtime »
Dépaquetage de vim-runtime (2:8.0.1453-1ubuntu1.1) ...
Sélection du paquet vim précédemment désélectionné.
Préparation du dépaquetage de .../vim_2:8.0.1453-1ubuntu1.1_amd64.deb ...
Dépaquetage de vim (2:8.0.1453-1ubuntu1.1) ...
Paramétrage de vim-runtime (2:8.0.1453-1ubuntu1.1) ...
Paramétrage de vim (2:8.0.1453-1ubuntu1.1) ...
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vim » (vim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vimdiff » (vimdiff) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rvim » (rvim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rview » (rview) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vi » (vi) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/view » (view) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/ex » (ex) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.8.3-2ubuntu0.1) ...

```

- Recherche du logiciel à installer.

Les actions d'un gestionnaire de paquets

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

```

root@Kevin-VirtualBox:/home/kevin# apt install vim
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  vim-runtime
Paquets suggérés :
  ctags vim-doc vim-scripts
Les MOUNGALK paquets suivants seront installés :
  vim vim-runtime
0 mis à jour, 2 nouvellement installés, 0 à enlever et 1 non mis à jour.
Il est nécessaire de prendre 0 507 ko dans les archives.
Après cette opération, 32,0 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] o
Réception de :1 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim-runtime all 2:8.0.1453-1ubuntu1.1 [5 435 kB]
Réception de :2 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim amd64 2:8.0.1453-1ubuntu1.1 [1 152 kB]
6 507 ko réceptionnés en 5s (1 437 ko/s)
Sélection du paquet vim-runtime précédemment désélectionné.
(Lecture de la base de données... 191012 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../vim-runtime_2:8.0.1453-1ubuntu1.1_all.deb ...
Ajout de « détournement de /usr/share/vim/vim80/doc/help.txt en /usr/share/vim/vim80/doc/help.txt.vim-tiny par vim-runtime »
Ajout de « détournement de /usr/share/vim/vim80/doc/tags en /usr/share/vim/vim80/doc/tags.vim-tiny par vim-runtime »
Dépaquetage de vim-runtime (2:8.0.1453-1ubuntu1.1) ...
Sélection du paquet vim précédemment désélectionné.
Préparation du dépaquetage de .../vim_2:8.0.1453-1ubuntu1.1_amd64.deb ...
Dépaquetage de vim (2:8.0.1453-1ubuntu1.1) ...
Paramétrage de vim-runtime (2:8.0.1453-1ubuntu1.1) ...
Paramétrage de vim (2:8.0.1453-1ubuntu1.1) ...
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vim » (vim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vimdiff » (vimdiff) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rvim » (rvim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rview » (rview) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vi » (vi) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/view » (view) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/ex » (ex) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.8.3-2ubuntu0.1) ...

```

- Recherche du logiciel à installer.
- Recherche des logiciels dont il dépend.

Les actions d'un gestionnaire de paquets

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

```

root@Kevin-VirtualBox:/home/kevin# apt install vim
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  vim-runtime
Paquets suggérés :
  ctags vim-doc vim-scripts
Les MOUNGAIK paquets suivants seront installés :
  vim-runtime
0 mis à jour, 2 nouvellement installés, 0 à enlever et 1 non mis à jour.
Il est nécessaire de prendre 0 507 ko dans les archives.
Après cette opération, 32,0 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] o
Réception de :1 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim-runtime all 2:8.0.1453-1ubuntu1.1 [5 435 kB]
Réception de :2 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim amd64 2:8.0.1453-1ubuntu1.1 [1 152 kB]
6 507 ko réceptionnés en 5s (1 437 ko/s)
Sélection du paquet vim-runtime précédemment désélectionné.
(Lecture de la base de données... 191012 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../vim-runtime_2:8.0.1453-1ubuntu1.1_all.deb ...
Ajout de « détournement de /usr/share/vim/vim80/doc/help.txt en /usr/share/vim/vim80/doc/help.txt.vim-tiny par vim-runtime »
Ajout de « détournement de /usr/share/vim/vim80/doc/tags en /usr/share/vim/vim80/doc/tags.vim-tiny par vim-runtime »
Dépaquetage de vim-runtime (2:8.0.1453-1ubuntu1.1) ...
Sélection du paquet vim précédemment désélectionné.
Préparation du dépaquetage de .../vim_2:8.0.1453-1ubuntu1.1_amd64.deb ...
Dépaquetage de vim (2:8.0.1453-1ubuntu1.1) ...
Paramétrage de vim-runtime (2:8.0.1453-1ubuntu1.1) ...
Paramétrage de vim (2:8.0.1453-1ubuntu1.1) ...
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vim » (vim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vimdiff » (vimdiff) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rvim » (rvim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rview » (rview) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vi » (vi) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/view » (view) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/ex » (ex) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.8.3-2ubuntu0.1) ...

```

- Recherche du logiciel à installer.
- Recherche des logiciels dont il dépend.
- Téléchargement automatique des paquets.

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les actions d'un gestionnaire de paquets

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

```

root@Kevin-VirtualBox:/home/kevin# apt install vim
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  vim-runtime
Paquets suggérés :
  ctags vim-doc vim-scripts
Les MOUNGALK paquets suivants seront installés :
  vim vim-runtime
0 mis à jour, 2 nouvellement installés, 0 à enlever et 1 non mis à jour.
Il est nécessaire de prendre 0 507 ko dans les archives.
Après cette opération, 32,0 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] o
Réception de :1 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim-runtime all 2:8.0.1453-1ubuntu1.1 [5 435 kB]
Réception de :2 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 vim amd64 2:8.0.1453-1ubuntu1.1 [1 152 kB]
6 507 ko réceptionnés en 5s (1 437 ko/s)
Sélection du paquet vim-runtime précédemment désélectionné.
(Lecture de la base de données... 191012 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../vim-runtime-2:8.0.1453-1ubuntu1.1.all.deb ...
Ajout de « détournement de /usr/share/vim/vim80/doc/help.txt en /usr/share/vim/vim80/doc/help.txt.vim-tiny par vim-runtime »
Ajout de « détournement de /usr/share/vim/vim80/doc/tags en /usr/share/vim/vim80/doc/tags.vim-tiny par vim-runtime »
Dépaquetage de vim-runtime (2:8.0.1453-1ubuntu1.1) ...
Sélection du paquet vim précédemment désélectionné.
Préparation du dépaquetage de .../vim-2:8.0.1453-1ubuntu1.1.amd64.deb ...
Dépaquetage de vim (2:8.0.1453-1ubuntu1.1) ...
Paramétrage de vim-runtime (2:8.0.1453-1ubuntu1.1) ...
Paramétrage de vim (2:8.0.1453-1ubuntu1.1) ...
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vim » (vim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vimdiff » (vimdiff) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rvim » (rvim) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/rview » (rview) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/vis » (vis) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/view » (view) en mode automatique
update-alternatives: utilisation de « /usr/bin/vim.basic » pour fournir « /usr/bin/ex » (ex) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.8.3-2ubuntu0.1) ...

```

- Recherche du logiciel à installer.
- Recherche des logiciels dont il dépend.
- Téléchargement automatique des paquets.
- Installations, configurations et tests des différents paquets.

Comprendre d'où viennent les paquets

Lecture du fichier `/etc/apt/sources.list`

```
root # cat /etc/apt/sources.list
```

```
deb http://fr.archive.ubuntu.com/ubuntu/ bionic main restricted
deb http://fr.archive.ubuntu.com/ubuntu/ bionic-updates main restricted
deb http://fr.archive.ubuntu.com/ubuntu/ bionic universe
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Comprendre d'où viennent les paquets

Lecture du fichier `/etc/apt/sources.list`

```
root # cat /etc/apt/sources.list
```

```
deb http://fr.archive.ubuntu.com/ubuntu/ bionic main restricted
deb http://fr.archive.ubuntu.com/ubuntu/ bionic-updates main restricted
deb http://fr.archive.ubuntu.com/ubuntu/ bionic universe
```

Figure – Ouverture de l'adresse d'un dépôt avec *Firefox*

← → ↻ 🏠 🔒 fr.archive.ubuntu.com/ubuntu/ubuntu/dists/bionic/universe/binary-amd64/

https://fr.archive.ubuntu.com

French archive server for Ubuntu and linux distributions based on Ubuntu.

Name	Last modified	Size
Parent Directory		-
by-hash/	2017-10-25 01:16	-
Packages.gz	2018-04-27 01:38	11M
Packages.xz	2018-04-27 01:38	8.2M
Release	2018-04-27 01:38	100

[Home](#) - [Recent updates](#) - [Stats day-1](#) - [Network statistics](#) - [Archive: dists folder](#)

Package archive: [Main](#) (free + supported) - [Restricted](#) (non-free + supported) - [Universe](#) (free + unsupported) - [Multiverse](#) (non-free + unsupported)

Information sur *Kapman* dans *Packages*

```

260076 Package: kapman
260077 Architecture: amd64
260078 Version: 4:17.12.3-0ubuntu1
260079 Priority: optional
260080 Section: universe/games
260081 Origin: Ubuntu
260082 Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
260083 Original-Maintainer: Debian/Kubuntu Qt/KDE Maintainers <debian-qt-kde@lists.debian.org>
260084 Bugs: https://bugs.launchpad.net/ubuntu/+filebug
260085 Installed-Size: 3237
260086 Depends: libc6 (>= 2.14), libkf5configcore5 (>= 4.98.0), libkf5configgui5 (>= 4.97.0), libkf5config7 (>= 4:14.12.50), libkf5kdegamesprivate1 (>= 4:14.12.50), libkf5widgetsaddons5 (>= 4.96.0), libkf5libqt5xml5 (>= 5.6.1~), libstdc++6 (>= 4.1.1)
260087 Recommends: khelpcenter
260088 Breaks: kde-l10n-ar (<< 4:17.03.90-0~), kde-l10n-ast (<< 4:17.03.90-0~), kde-l10n-bg (<< 4:17.03.90-0~), kde-l10n-da (<< 4:17.03.90-0~), kde-l10n-de (<< 4:17.03.90-0~), kde-l10n-el (<< 4:17.03.90-0~), kde-l10n-en (<< 4:17.03.90-0~), kde-l10n-es (<< 4:17.03.90-0~), kde-l10n-fa (<< 4:17.03.90-0~), kde-l10n-fi (<< 4:17.03.90-0~), kde-l10n-fr (<< 4:17.03.90-0~), kde-l10n-ga (<< 4:17.03.90-0~), kde-l10n-gl (<< 4:17.03.90-0~), kde-l10n-he (<< 4:17.03.90-0~), kde-l10n-hi (<< 4:17.03.90-0~), kde-l10n-hr (<< 4:17.03.90-0~), kde-l10n-hu (<< 4:17.03.90-0~), kde-l10n-ia (<< 4:17.03.90-0~), kde-l10n-id (<< 4:17.03.90-0~), kde-l10n-it (<< 4:17.03.90-0~), kde-l10n-ja (<< 4:17.03.90-0~), kde-l10n-ka (<< 4:17.03.90-0~), kde-l10n-kk (<< 4:17.03.90-0~), kde-l10n-km (<< 4:17.03.90-0~), kde-l10n-ko (<< 4:17.03.90-0~), kde-l10n-lt (<< 4:17.03.90-0~), kde-l10n-lv (<< 4:17.03.90-0~), kde-l10n-nb (<< 4:17.03.90-0~), kde-l10n-nn (<< 4:17.03.90-0~), kde-l10n-pa (<< 4:17.03.90-0~), kde-l10n-pl (<< 4:17.03.90-0~), kde-l10n-pt (<< 4:17.03.90-0~), kde-l10n-pt_BR (<< 4:17.03.90-0~), kde-l10n-ro (<< 4:17.03.90-0~), kde-l10n-ru (<< 4:17.03.90-0~), kde-l10n-si (<< 4:17.03.90-0~), kde-l10n-sl (<< 4:17.03.90-0~), kde-l10n-sr (<< 4:17.03.90-0~), kde-l10n-sv (<< 4:17.03.90-0~), kde-l10n-ta (<< 4:17.03.90-0~), kde-l10n-telugu (<< 4:17.03.90-0~), kde-l10n-th (<< 4:17.03.90-0~), kde-l10n-tk (<< 4:17.03.90-0~), kde-l10n-tr (<< 4:17.03.90-0~), kde-l10n-uk (<< 4:17.03.90-0~), kde-l10n-ur (<< 4:17.03.90-0~), kde-l10n-uz (<< 4:17.03.90-0~), kde-l10n-vi (<< 4:17.03.90-0~), kde-l10n-xh (<< 4:17.03.90-0~), kde-l10n-zh (<< 4:17.03.90-0~), kde-l10n-zh_CN (<< 4:17.03.90-0~), kde-l10n-zh_TW (<< 4:17.03.90-0~)
260089 Replaces: kde-l10n-ar (<< 4:17.03.90-0~), kde-l10n-ast (<< 4:17.03.90-0~), kde-l10n-bg (<< 4:17.03.90-0~), kde-l10n-da (<< 4:17.03.90-0~), kde-l10n-de (<< 4:17.03.90-0~), kde-l10n-el (<< 4:17.03.90-0~), kde-l10n-en (<< 4:17.03.90-0~), kde-l10n-es (<< 4:17.03.90-0~), kde-l10n-fa (<< 4:17.03.90-0~), kde-l10n-fi (<< 4:17.03.90-0~), kde-l10n-fr (<< 4:17.03.90-0~), kde-l10n-ga (<< 4:17.03.90-0~), kde-l10n-gl (<< 4:17.03.90-0~), kde-l10n-he (<< 4:17.03.90-0~), kde-l10n-hi (<< 4:17.03.90-0~), kde-l10n-hr (<< 4:17.03.90-0~), kde-l10n-hu (<< 4:17.03.90-0~), kde-l10n-ia (<< 4:17.03.90-0~), kde-l10n-id (<< 4:17.03.90-0~), kde-l10n-it (<< 4:17.03.90-0~), kde-l10n-ja (<< 4:17.03.90-0~), kde-l10n-ka (<< 4:17.03.90-0~), kde-l10n-kk (<< 4:17.03.90-0~), kde-l10n-km (<< 4:17.03.90-0~), kde-l10n-ko (<< 4:17.03.90-0~), kde-l10n-lt (<< 4:17.03.90-0~), kde-l10n-lv (<< 4:17.03.90-0~), kde-l10n-nb (<< 4:17.03.90-0~), kde-l10n-nn (<< 4:17.03.90-0~), kde-l10n-pa (<< 4:17.03.90-0~), kde-l10n-pl (<< 4:17.03.90-0~), kde-l10n-pt (<< 4:17.03.90-0~), kde-l10n-pt_BR (<< 4:17.03.90-0~), kde-l10n-ro (<< 4:17.03.90-0~), kde-l10n-ru (<< 4:17.03.90-0~), kde-l10n-si (<< 4:17.03.90-0~), kde-l10n-sl (<< 4:17.03.90-0~), kde-l10n-sr (<< 4:17.03.90-0~), kde-l10n-sv (<< 4:17.03.90-0~), kde-l10n-ta (<< 4:17.03.90-0~), kde-l10n-telugu (<< 4:17.03.90-0~), kde-l10n-th (<< 4:17.03.90-0~), kde-l10n-tk (<< 4:17.03.90-0~), kde-l10n-tr (<< 4:17.03.90-0~), kde-l10n-uk (<< 4:17.03.90-0~), kde-l10n-ur (<< 4:17.03.90-0~), kde-l10n-uz (<< 4:17.03.90-0~), kde-l10n-vi (<< 4:17.03.90-0~), kde-l10n-xh (<< 4:17.03.90-0~), kde-l10n-zh (<< 4:17.03.90-0~), kde-l10n-zh_CN (<< 4:17.03.90-0~), kde-l10n-zh_TW (<< 4:17.03.90-0~)
260090 Filename: pool/universe/k/kapman/kapman_17.12.3-0ubuntu1_amd64.deb
260091 Size: 2595696
260092 MD5sum: ff5271bd165dc0ede46819de69fd743c
260093 SHA1: 8f408577bcd6868d03d3c1b5a39fa4a890c3fff1
260094 SHA256: 733dcedd85193df37ebe0ae3682abf4fda811de1136092c1bfe839ad1bca0618
260095 Homepage: http://games.kde.org/
260096 Description: Pac-Man clone
260097 Description-md5: 1d94f103a6072e04515d6984872e0b4b
260098 Supported: 3y

```

Ajout d'un nouveau dépôt *APT*

Ajout d'une nouvelle clé *GPG* pouvant signer les paquets

```
root # curl -sS https://download.spotify.com/debian/pubkey.gpg |  
sudo apt-key add -  
OK
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

**Gestion des
logiciels**

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Ajout d'un nouveau dépôt *APT*

Ajout d'une nouvelle clé *GPG* pouvant signer les paquets

```
root # curl -sS https://download.spotify.com/debian/pubkey.gpg |  
sudo apt-key add -  
OK
```

Ajout d'un nouveau dépôt sur lequel rechercher les logiciels

```
root # echo "deb http://repository.spotify.com stable non-free" |  
sudo tee /etc/apt/sources.list.d/spotify.list
```

Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs
- Les sauvegardes

Conclusion

Ajout d'un nouveau dépôt *APT*Ajout d'une nouvelle clé *GPG* pouvant signer les paquets

```
root # curl -sS https://download.spotify.com/debian/pubkey.gpg |  
sudo apt-key add -  
OK
```

Ajout d'un nouveau dépôt sur lequel rechercher les logiciels

```
root # echo "deb http://repository.spotify.com stable non-free" |  
sudo tee /etc/apt/sources.list.d/spotify.list
```

Lecture du nouveau fichier *spotify.list*

```
kevin $ cat /etc/apt/sources.list.d/spotify.list  
deb http://repository.spotify.com stable non-free
```

Installation de *Spotify*

Mise à jour de la liste des paquets

```
root # apt update
```

```
Réception de :12 http://repository.spotify.com stable/non-free i386  
Packages [1 171 B]
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

**Gestion des
logiciels**

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Mise à jour de la liste des paquets

```
root # apt update
```

```
Réception de :12 http://repository.spotify.com stable/non-free i386  
Packages [1 171 B]
```

Recherche d'un paquet *Spotify*

```
root # apt search spotify
```

```
spotify-client/stable 1:1.1.10.546.ge08ef575-19 amd64  
Spotify streaming music client
```

Mise à jour de la liste des paquets

```
root # apt update
```

```
Réception de :12 http://repository.spotify.com stable/non-free i386  
Packages [1 171 B]
```

Recherche d'un paquet *Spotify*

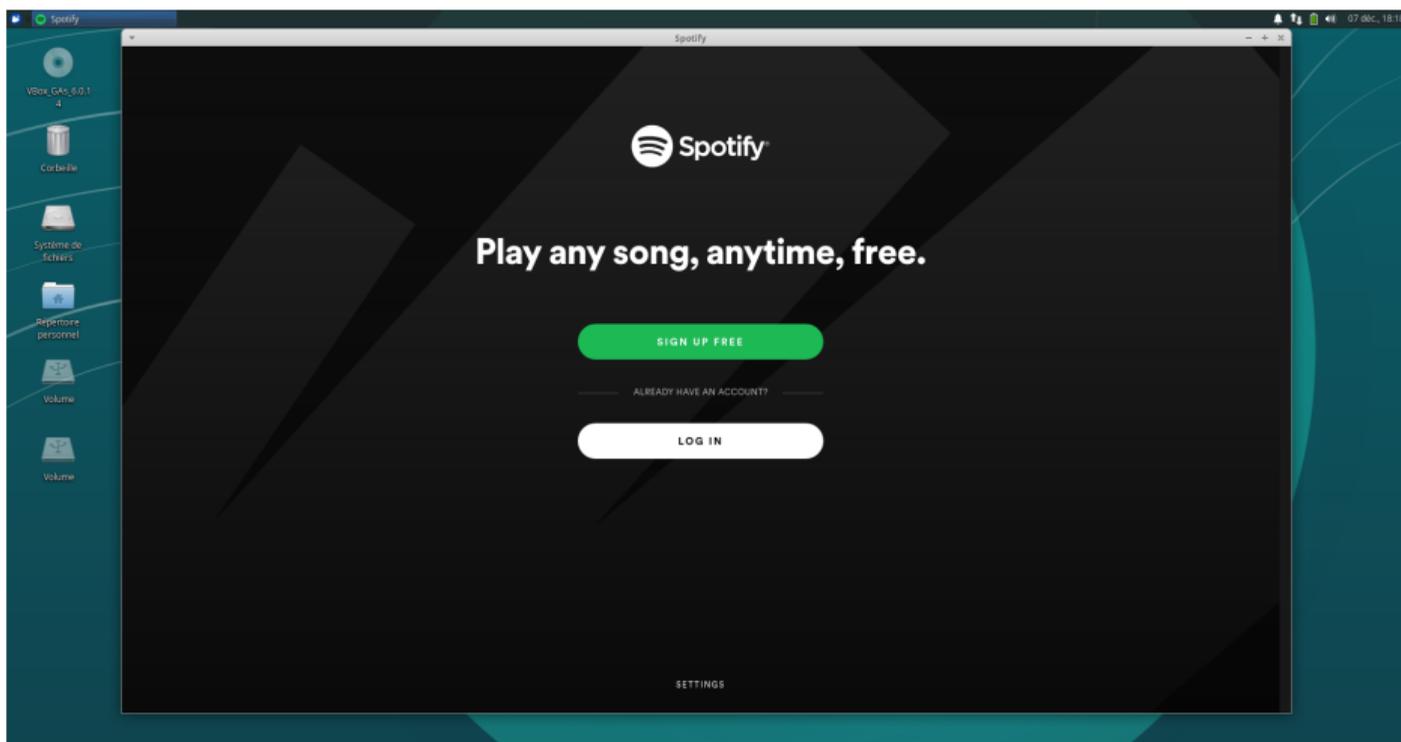
```
root # apt search spotify
```

```
spotify-client/stable 1:1.1.10.546.ge08ef575-19 amd64  
Spotify streaming music client
```

Installation de *spotify-client*

```
root # apt install spotify-client
```

Spotify est bien installé !



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

**Gestion des
logiciels**

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Le noyau *Linux* n'est pas grand-chose d'utilisable en tant que tel. C'est l'écosystème gravitant avec lui qui en fait sa force.
- Il est généralement présent au sein d'une *distribution*.
- *Xubuntu* est une distribution *Linux* grand public basée sur le gestionnaire de paquets *APT*.
- Le gestionnaire de paquets permet de s'assurer de la bonne récupération d'un logiciel, de ses dépendances et de la génération d'une partie de sa configuration.
- En plus des développeurs d'une application, beaucoup de personnes travaillent dur au niveau des distributions pour faciliter l'intégration des logiciels au sein de l'écosystème déjà existant.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 3.1

GNU

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Richard Stallman



- Né le 16 mars 1953 à *Manhattan*.
- Lance le projet *GNU* en 1983.
- Fonde la *Free Software Foundation (FSF)* en 1985.
- Développe les logiciels *Emacs*, *GCC* et *gdb*.
- Militant très actif de la philosophie du libre.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- En 1983, Richard annonce l'implémentation d'un système d'exploitation équivalent à *UNIX*, mais en libre. Il s'agit du projet *GNU* : *GNU is Not Unix*.
- En 1990, la plupart des éléments du système *GNU* sont prêts. . . à l'exception du noyau. Le développement de *Hurd* est donc initialisé, mais celui-ci se révélera beaucoup plus long que prévu.
- En 1991, *Linus* crée *Linux* et y adapte les outils *GNU* de façon à obtenir très vite quelque chose de réellement utilisable. *GNU/Linux* était né.
- Aujourd'hui, il existe une multitude de logiciels *GNU* dont la plupart ne sont pas nécessaires pour le bon fonctionnement de *Linux* comme *mediagoblin*, *Emacs*, ou *Dia*. Une liste complète est disponible [ici](#)¹³.

Logiciels *GNU* importants

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Logiciels *GNU* importants

- Bash
- Binutils
- Coreutils
- Gcc / Make
- Glibc

GNU General Public License (*GNU GPL*)

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

GNU General Public License (*GNU GPL*)

- Les quatre libertés de l'utilisateur :
 - La liberté d'utiliser le logiciel pour n'importe quel but.
 - La liberté de partager le logiciel à d'autres.
 - La liberté de modifier le logiciel en fonction de ses besoins.
 - La liberté de partager ces modifications avec tout le monde.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

GNU General Public License (*GNU GPL*)

- Les quatre libertés de l'utilisateur :
 - La liberté d'utiliser le logiciel pour n'importe quel but.
 - La liberté de partager le logiciel à d'autres.
 - La liberté de modifier le logiciel en fonction de ses besoins.
 - La liberté de partager ces modifications avec tout le monde.
- Le copyleft : plutôt que d'utiliser le droit d'auteur pour restreindre l'accès à une œuvre, il est utilisé pour en garantir sa liberté.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

GNU General Public License (*GNU GPL*)

- Les quatre libertés de l'utilisateur :
 - La liberté d'utiliser le logiciel pour n'importe quel but.
 - La liberté de partager le logiciel à d'autres.
 - La liberté de modifier le logiciel en fonction de ses besoins.
 - La liberté de partager ces modifications avec tout le monde.
- Le copyleft : plutôt que d'utiliser le droit d'auteur pour restreindre l'accès à une œuvre, il est utilisé pour en garantir sa liberté.
- « Libre » ne veut pas dire « gratuit » !

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

GNU General Public License (*GNU GPL*)

- Les quatre libertés de l'utilisateur :
 - La liberté d'utiliser le logiciel pour n'importe quel but.
 - La liberté de partager le logiciel à d'autres.
 - La liberté de modifier le logiciel en fonction de ses besoins.
 - La liberté de partager ces modifications avec tout le monde.
- Le copyleft : plutôt que d'utiliser le droit d'auteur pour restreindre l'accès à une œuvre, il est utilisé pour en garantir sa liberté.
- « Libre » ne veut pas dire « gratuit » !
- Un logiciel sous *GPL* le sera toujours, une autre application l'utilisant devra être distribuée avec une licence similaire.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

GNU General Public License (*GNU GPL*)

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Les quatre libertés de l'utilisateur :
 - La liberté d'utiliser le logiciel pour n'importe quel but.
 - La liberté de partager le logiciel à d'autres.
 - La liberté de modifier le logiciel en fonction de ses besoins.
 - La liberté de partager ces modifications avec tout le monde.
- Le copyleft : plutôt que d'utiliser le droit d'auteur pour restreindre l'accès à une œuvre, il est utilisé pour en garantir sa liberté.
- « Libre » ne veut pas dire « gratuit » !
- Un logiciel sous *GPL* le sera toujours, une autre application l'utilisant devra être distribuée avec une licence similaire.
- Deux variantes existent :
 - *LGPL*

GNU General Public License (*GNU GPL*)

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Les quatre libertés de l'utilisateur :
 - La liberté d'utiliser le logiciel pour n'importe quel but.
 - La liberté de partager le logiciel à d'autres.
 - La liberté de modifier le logiciel en fonction de ses besoins.
 - La liberté de partager ces modifications avec tout le monde.
- Le copyleft : plutôt que d'utiliser le droit d'auteur pour restreindre l'accès à une œuvre, il est utilisé pour en garantir sa liberté.
- « Libre » ne veut pas dire « gratuit » !
- Un logiciel sous *GPL* le sera toujours, une autre application l'utilisant devra être distribuée avec une licence similaire.
- Deux variantes existent :
 - *LGPL*, pour les bibliothèques non compétitives.
 - *AGPL*

GNU General Public License (*GNU GPL*)

- Les quatre libertés de l'utilisateur :
 - La liberté d'utiliser le logiciel pour n'importe quel but.
 - La liberté de partager le logiciel à d'autres.
 - La liberté de modifier le logiciel en fonction de ses besoins.
 - La liberté de partager ces modifications avec tout le monde.
- Le copyleft : plutôt que d'utiliser le droit d'auteur pour restreindre l'accès à une œuvre, il est utilisé pour en garantir sa liberté.
- « Libre » ne veut pas dire « gratuit » !
- Un logiciel sous *GPL* le sera toujours, une autre application l'utilisant devra être distribuée avec une licence similaire.
- Deux variantes existent :
 - *LGPL*, pour les bibliothèques non compétitives.
 - *AGPL*, pour les logiciels de serveurs.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Des alternatives

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- *BusyBox*, une réécriture des binaires disponibles dans les paquets *GNU* : *fileutils* et *shellutils*. Ces versions sont moins complètes que leurs équivalents *GNU* mais sont bien plus légères.
- *Utils*, une réécriture de *GNU coreutils* en *Rust*.
- *Musl*, une réécriture de *GNU glibc* censée être plus légère et optimisée.
- *Zsh*, *Ash*, etc. Une multitude de shells autre que *GNU Bash* existent.

Exercice 3.2

Le shell

Le shell (ou « coque logicielle », en français)

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

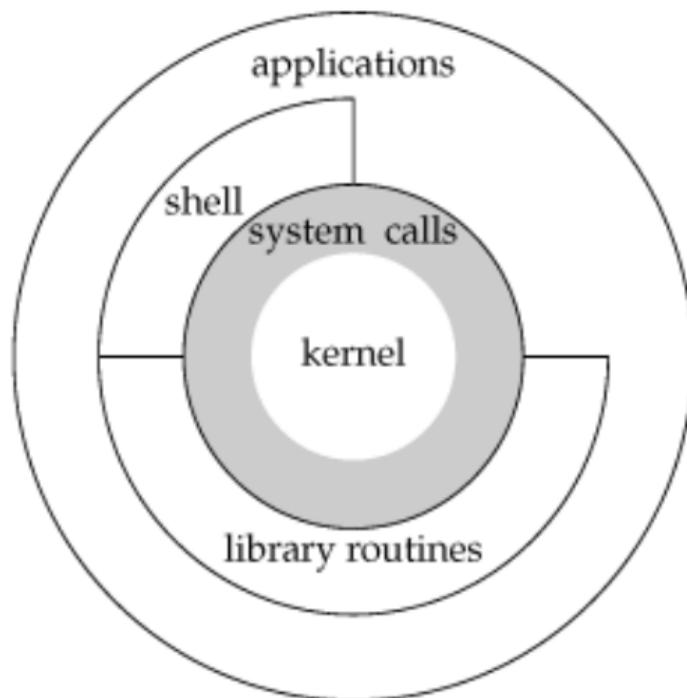
Btrfs

Les sauvegardes

Conclusion

Le shell (ou « coque logicielle », en français)

Figure – Une vue d'ensemble (*crédit*^[4])



Deux principaux types de shells

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

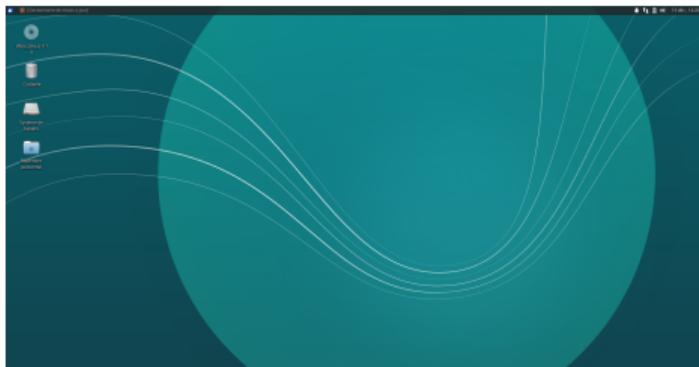
Le stockage

Btrfs

Les sauvegardes

Conclusion

Deux principaux types de shells



Les *GUI* (Graphical User Interface)

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

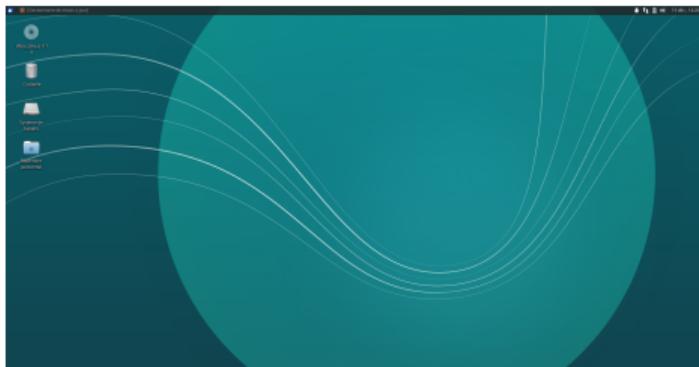
Le stockage

Btrfs

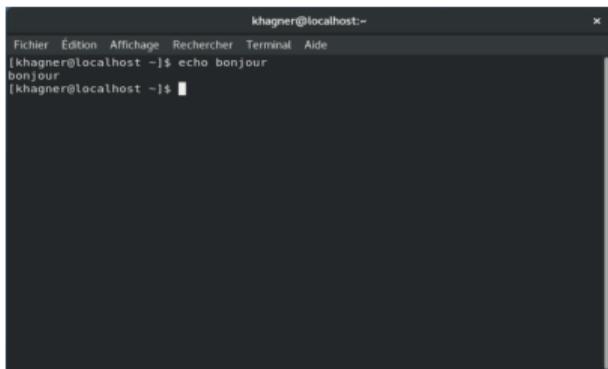
Les sauvegardes

Conclusion

Deux principaux types de shells



Les *GUI* (Graphical User Interface)



Les *CLI* (Command Line Interface)

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

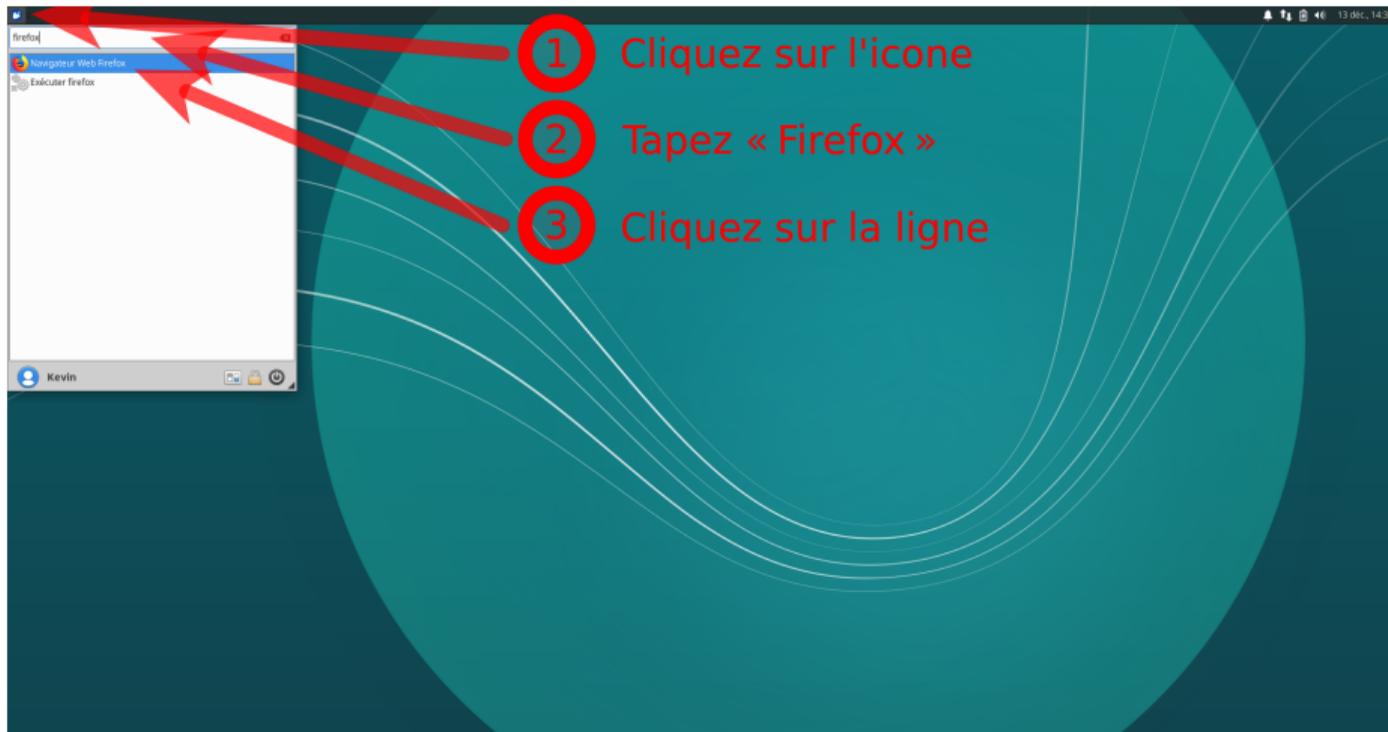
Le stockage

Btrfs

Les sauvegardes

Conclusion

Lancer *Firefox* depuis *XFCE*

Lancer *Firefox* depuis *XFCE*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Lancer *Firefox* depuis *Bash*

Lancer *Firefox* depuis *Bash*

Rappel : *Bash* est le shell *GNU* présent dans presque toutes les distributions *Linux*.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Lancer *Firefox* depuis *Bash*

Rappel : *Bash* est le shell *GNU* présent dans presque toutes les distributions *Linux*.

Exécution du binaire *firefox* à l'aide de `$PATH`

```
kevin $ firefox
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Lancer *Firefox* depuis *Bash*

Rappel : *Bash* est le shell *GNU* présent dans presque toutes les distributions *Linux*.

Exécution du binaire *firefox* à l'aide de `$PATH`

```
kevin $ firefox
```

Localisation du binaire *firefox*

```
kevin $ which firefox  
/usr/bin/firefox
```

Lancer *Firefox* depuis *Bash*

Rappel : *Bash* est le shell *GNU* présent dans presque toutes les distributions *Linux*.

Exécution du binaire *firefox* à l'aide de `$PATH`

```
kevin $ firefox
```

Localisation du binaire *firefox*

```
kevin $ which firefox  
/usr/bin/firefox
```

Exécution du binaire *firefox* depuis son chemin absolu

```
kevin $ /usr/bin/firefox
```

Les variables d'environnement

- C'est quoi ?

Les variables d'environnement

- C'est quoi ?
Options permettant d'influencer le comportement de logiciels. Exemple : `$LANG` permet de définir le langage préféré d'un utilisateur. Une application pourra donc utiliser cette variable pour savoir en quelle langue afficher ses messages.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les variables d'environnement

- C'est quoi ?
Options permettant d'influencer le comportement de logiciels. Exemple : `$LANG` permet de définir le langage préféré d'un utilisateur. Une application pourra donc utiliser cette variable pour savoir en quelle langue afficher ses messages.
- Comment afficher la variable contenant le nom de l'utilisateur courant ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les variables d'environnement

- C'est quoi ?
Options permettant d'influencer le comportement de logiciels. Exemple : \$LANG permet de définir le langage préféré d'un utilisateur. Une application pourra donc utiliser cette variable pour savoir en quelle langue afficher ses messages.
- Comment afficher la variable contenant le nom de l'utilisateur courant ?

```
Affichage du contenu de la variable $USER
kevin $ echo $USER
kevin
```

- À quoi sert \$PATH ?

Les variables d'environnement

- C'est quoi ?

Options permettant d'influencer le comportement de logiciels. Exemple : `$LANG` permet de définir le langage préféré d'un utilisateur. Une application pourra donc utiliser cette variable pour savoir en quelle langue afficher ses messages.

- Comment afficher la variable contenant le nom de l'utilisateur courant ?

Affichage du contenu de la variable `$USER`

```
kevin $ echo $USER
kevin
```

- À quoi sert `$PATH` ?

Affichage du contenu de la variable `$PATH`

```
kevin $ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

Utilisé par *Bash* pour savoir où chercher les binaires appelés de façon implicite.

Exercice 3.3

Le shell : de l'aide et des options

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Permet d'afficher les pages de manuel présentes dans `/usr/share/man`.
- Par convention, tous les logiciels possèdent des pages de manuels.
- Le caractère slash (`/`) permet de faire des recherches, `n` permet de passer à l'occurrence suivante.
- `man man`
- `man firefox`

/usr/share/info

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau
UNIX
Linux
Les distributions
Conclusion

Installation

Xubuntu
Les mises à jour
Gestion des logiciels

Les bases

GNU
Le shell
Les utilisateurs
Les fichiers
Les processus
Les scripts

Les données

Le stockage
Btrfs
Les sauvegardes

Conclusion

- Contient d'autres documentations, mais dans des formats plus variés que ce qu'on trouve dans `/usr/share/man`. Ces docs sont aussi généralement plus complètes (si elles existent!).
- La commande `info` permet de lire certains fichiers contenus à cet emplacement.
- Ces documentations peuvent y être en format *info*, *html*, voir en d'autres formats encore.
- La seule façon fiable de savoir si une documentation existe pour un logiciel en particulier à cet emplacement et d'aller voir manuellement.

Options : effectuer une capture d'écran avec *Firefox*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Options : effectuer une capture d'écran avec *Firefox*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Affichage de l'aide de *Firefox* grâce à l'option `-h`

```
kevin $ firefox -h
```

```
Usage: firefox [ options ... ] [URL]
```

```
where options include:
```

```
Firefox options
```

```
--screenshot [<path>] Save screenshot to <path> or in working  
directory.
```

Options : effectuer une capture d'écran avec *Firefox*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Affichage de l'aide de *Firefox* grâce à l'option `-h`

```
kevin $ firefox -h
```

```
Usage: firefox [ options ... ] [URL]
```

```
where options include:
```

```
Firefox options
```

```
--screenshot [<path>] Save screenshot to <path> or in working  
directory.
```

Création d'un screenshot avec *Firefox* de la page *cesi.fr*

```
kevin $ firefox --screenshot /tmp/cesi.png http://cesi.fr
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Liste des logiciels contenant « *browser* » dans leurs descriptions courtes

```
kevin $ man -k browser
```

```
epiphany (1) - simple to use web browser for GNOME
```

```
firefox (1) - a Web browser for X11 derived from the Mozilla  
browser
```

```
git-web--browse (1) - Git helper script to launch a web browser
```

```
pinfo (1) - curses based lynx-style info browser
```

```
regshell (1) - Windows registry file browser using readline
```

```
smbtree (1) - A text based smb network browser
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Combinaison des options

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Combinaison des options

Affichage de la page de manuel de *ls*

kevin \$ man ls

NAME

ls - list directory contents

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Affichage de la page de manuel de *ls*

```
kevin $ man ls
```

```
NAME
```

```
ls - list directory contents
```

Lister le répertoire */var/log*

Afficher également la taille des fichiers

La taille à l'échelle optimale

Triés de façon croissant

Affichage de la page de manuel de *ls*

```
kevin $ man ls
```

```
NAME
```

```
ls - list directory contents
```

Lister le répertoire */var/log*

```
ls /var/log
```

Afficher également la taille des fichiers

La taille à l'échelle optimale

Triés de façon croissant

Affichage de la page de manuel de *ls*

```
kevin $ man ls
```

```
NAME
```

```
ls - list directory contents
```

Lister le répertoire */var/log*

```
ls /var/log
```

Afficher également la taille des fichiers

```
ls /var/log -s
```

La taille à l'échelle optimale

Triés de façon croissant

Affichage de la page de manuel de *ls*

```
kevin $ man ls
```

```
NAME
```

```
ls - list directory contents
```

Lister le répertoire */var/log*

```
ls /var/log
```

Afficher également la taille des fichiers

```
ls /var/log -s
```

La taille à l'échelle optimale

```
ls /var/log -sh
```

Triés de façon croissant

Combinaison des options

Affichage de la page de manuel de *ls*

```
kevin $ man ls
```

```
NAME
```

```
ls - list directory contents
```

Lister le répertoire */var/log*

```
ls /var/log
```

Afficher également la taille des fichiers

```
ls /var/log -s
```

La taille à l'échelle optimale

```
ls /var/log -sh
```

Triés de façon croissant

```
ls /var/log -shSr
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 3.4

L'enchaînement

La philosophie UNIX

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- « keep it simple, stupid » (*KISS*), un logiciel ne devrait faire seule chose et le faire de façon simple.
- Les logiciels doivent être interopérables.
- Les logiciels doivent pouvoir communiquer ensemble et ce flux doit être compréhensible par les humains.
- Prolonger avec l'article de [0x2501](#).

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les flux standards

Figure – Les trois flux (*crédit*[↗])

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

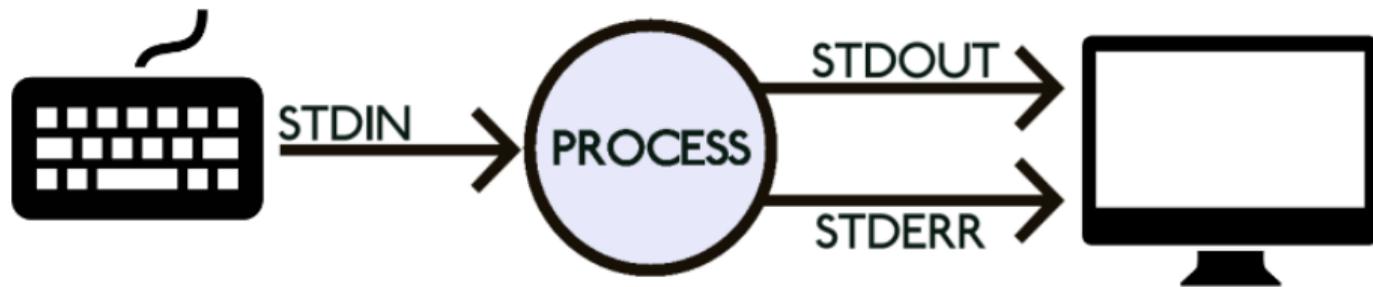
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Figure – Les trois flux (*crédit*[↗])

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

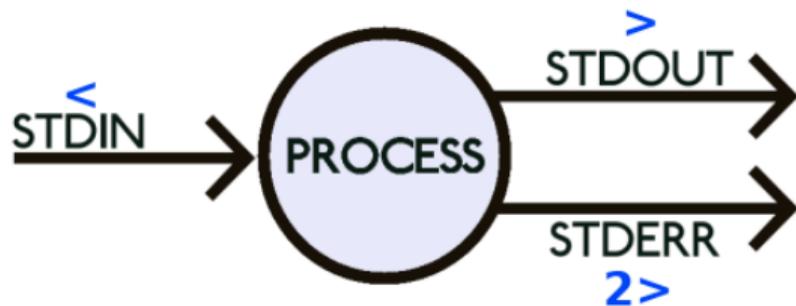
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Figure – Les trois flux (*crédit*^[2])

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La commande *cat*

Affiche la page de manuel de la commande *cat*

```
kevin $ man cat
```

```
CAT(1) User Commands CAT(1)
```

NAME

```
cat - concatenate files and print on the standard output
```

SYNOPSIS

```
cat [OPTION]... [FILE]...
```

DESCRIPTION

```
Concatenate FILE(s) to standard output.
```

```
With no FILE, or when FILE is -, read standard input.
```

STDIN – Défini le flux d'entrée du logiciel

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

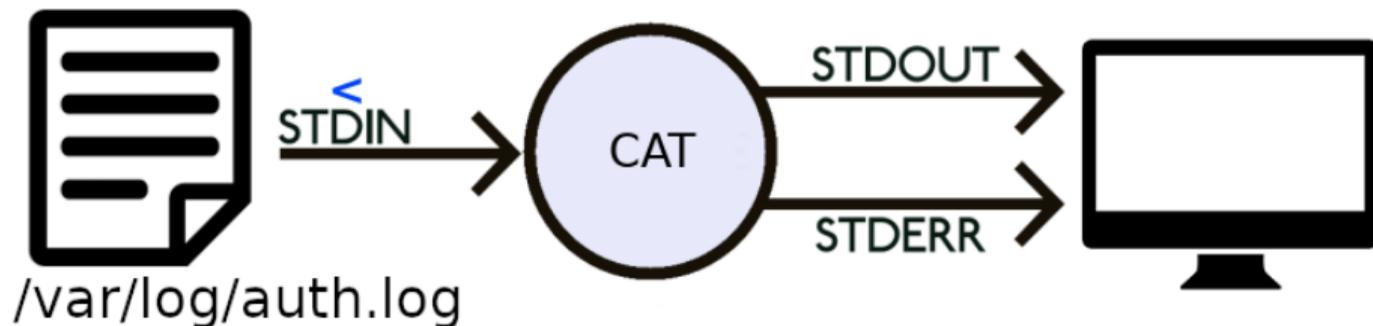
Les sauvegardes

Conclusion

STDIN – Défini le flux d'entrée du logiciel

Liste le contenu du fichier /var/log/auth.log

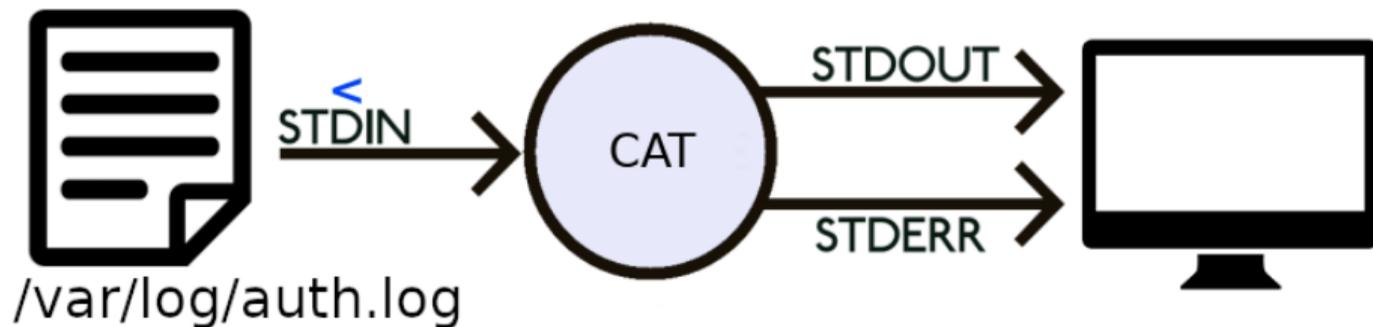
kevin \$

Figure – Redirection de *STDIN* vers un fichier

STDIN – Défini le flux d'entrée du logiciel

Liste le contenu du fichier /var/log/auth.log

```
kevin $ cat < /var/log/auth.log
```

Figure – Redirection de *STDIN* vers un fichier

STDOUT – Défini la sortie standard

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

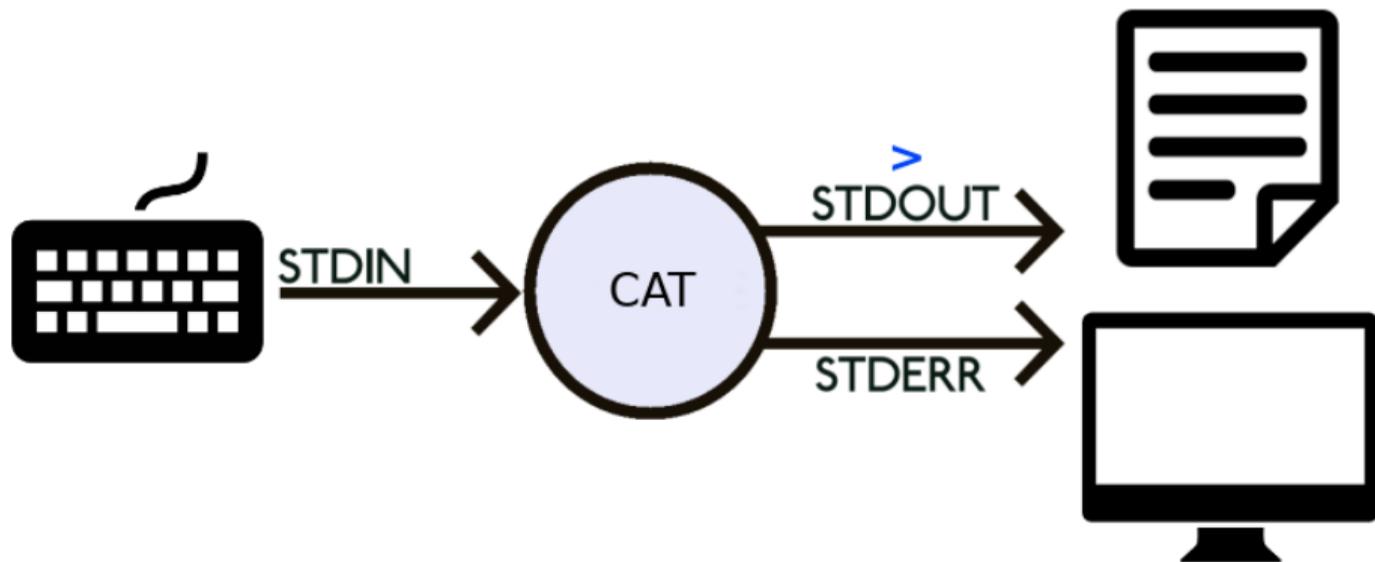
Conclusion

STDOUT – Défini la sortie standard

Écrit les frappes au clavier dans le fichier /tmp/fichier.txt

```
kevin $
```

Figure – Redirection de *STDOUT* vers un fichier

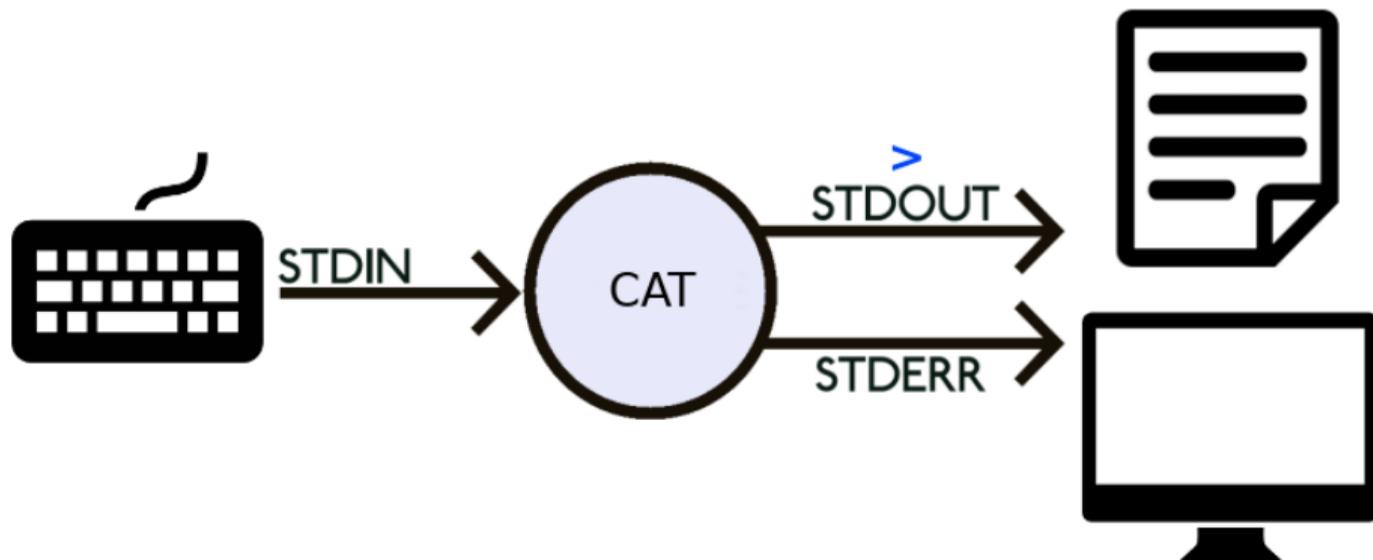


STDOUT – Défini la sortie standard

Écrit les frappes au clavier dans le fichier /tmp/fichier.txt

```
kevin $ cat > /tmp/fichier.txt
```

Figure – Redirection de *STDOUT* vers un fichier



STDERR – Défini la sortie d'erreur

Comment faire pour ne pas afficher les messages d'erreur retournés par la commande : `du /var/log?`

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

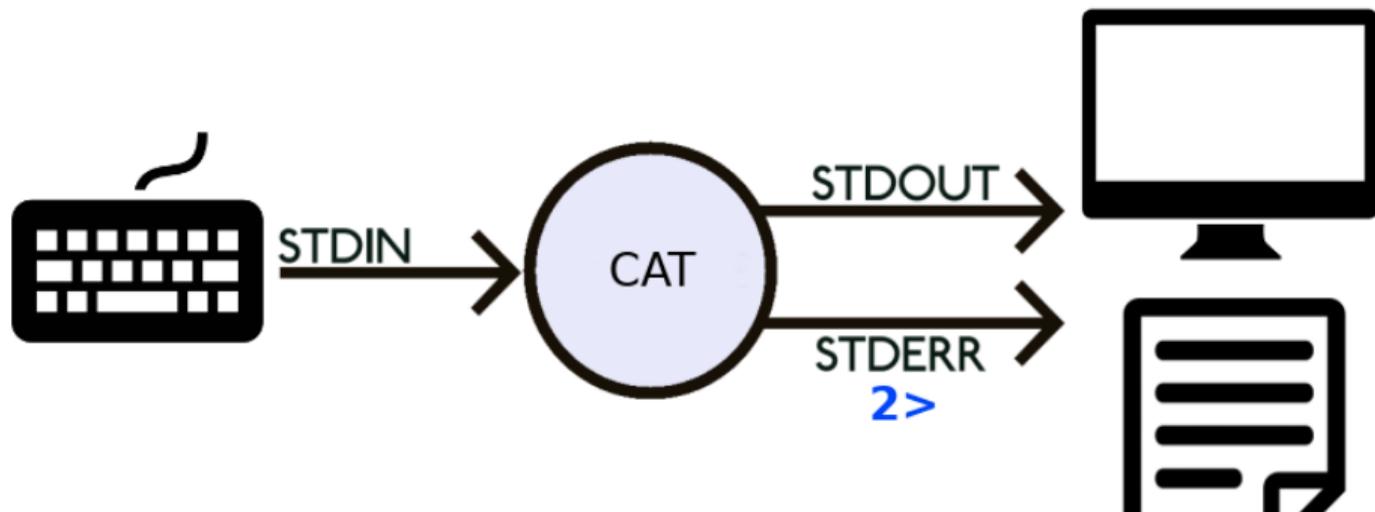
Les sauvegardes

Conclusion

STDERR – Défini la sortie d'erreur

Comment faire pour ne pas afficher les messages d'erreur retournés par la commande : du /var/log?

Figure – Redirection de *STDERR* vers un fichier



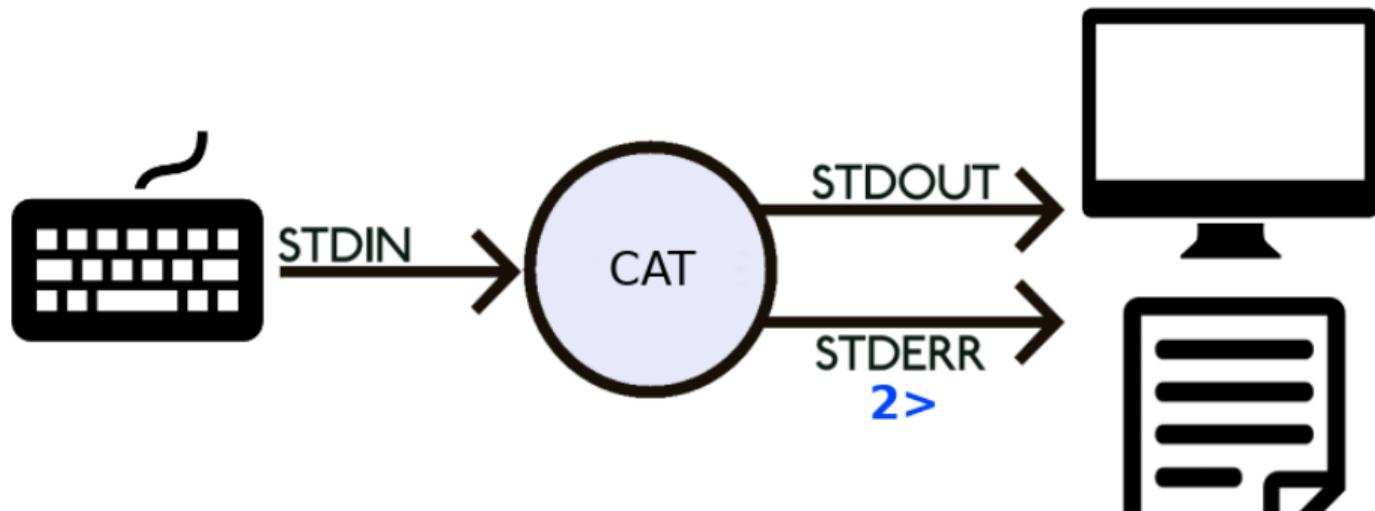
STDERR – Défini la sortie d'erreur

Comment faire pour ne pas afficher les messages d'erreur retournés par la commande : `du /var/log` ?

Redirection des erreurs de `du` vers `/dev/null`

```
kevin $ du /var/log 2> /dev/null
```

Figure – Redirection de *STDERR* vers un fichier



Comment faire pour utiliser `cat` pour copier un fichier `/tmp/entrée.txt` dans un fichier `/tmp/sortie.txt` en écrivant les éventuelles erreurs dans `/tmp/erreur.txt` ?

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Comment faire pour utiliser `cat` pour copier un fichier `/tmp/entrée.txt` dans un fichier `/tmp/sortie.txt` en écrivant les éventuelles erreurs dans `/tmp/erreur.txt` ?

Copie du fichier `/tmp/entrée.txt` à l'emplacement `/tmp/sortie.txt` en écrivant les erreurs dans `/tmp/erreur.txt`

```
kevin $ cat 2> /tmp/erreur.txt < /tmp/entrée.txt > /tmp/sortie.txt
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La commande *grep*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Manuel de la commande *grep*

```
kevin $ man grep
```

NAME

grep, egrep, fgrep - print lines that match patterns

SYNOPSIS

```
grep [OPTION...] PATTERNS [FILE...]
```

DESCRIPTION

grep searches for PATTERNS in each FILE. PATTERNS is one or patterns separated by newline characters, and grep prints each line that matches a pattern.

A FILE of “-” stands for standard input. If no FILE is given, recursive searches examine the working directory, and nonrecursive searches read standard input.

N'affiche que les lignes du fichier SUPPORTED contenant le mot « *fr* ».

```
kevin $
```

Manuel de la commande *grep*

```
kevin $ man grep
```

NAME

grep, egrep, fgrep - print lines that match patterns

SYNOPSIS

```
grep [OPTION...] PATTERNS [FILE...]
```

DESCRIPTION

grep searches for PATTERNS in each FILE. PATTERNS is one or patterns separated by newline characters, and grep prints each line that matches a pattern.

A FILE of “-” stands for standard input. If no FILE is given, recursive searches examine the working directory, and nonrecursive searches read standard input.

N'affiche que les lignes du fichier SUPPORTED contenant le mot « *fr* ».

```
kevin $ grep fr /usr/share/i18n/SUPPORTED
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

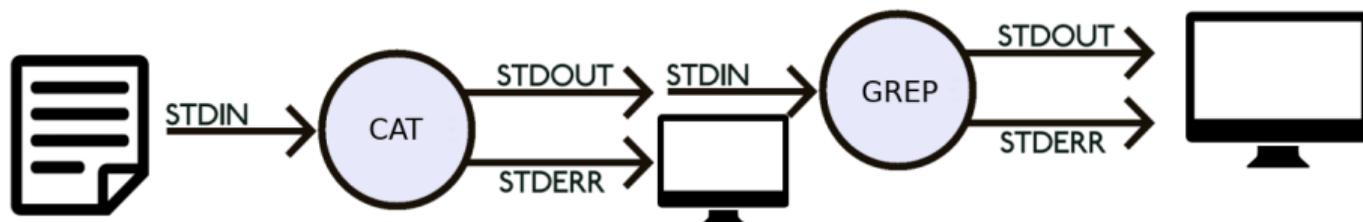
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Figure – Mettre un tuyau entre *cat* et *grep*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

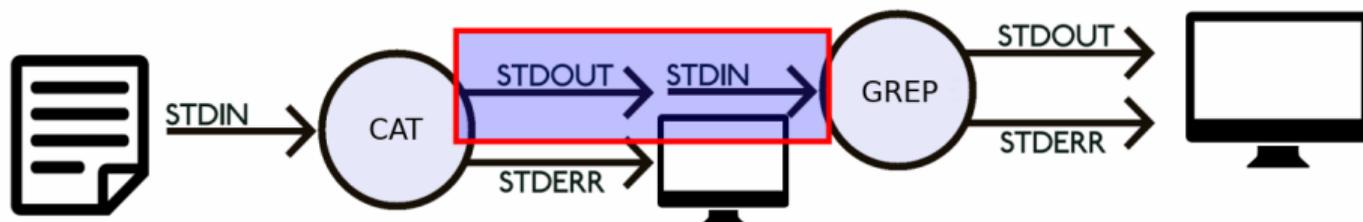
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Figure – Mettre un tuyau entre *cat* et *grep*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

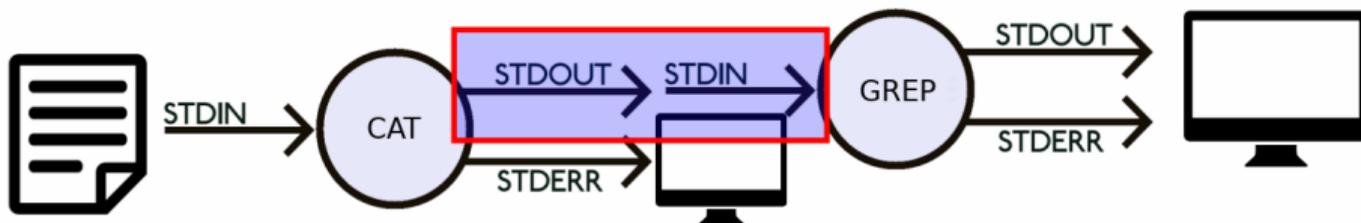
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Figure – Mettre un tuyau entre *cat* et *grep*Affichage des lignes du fichier SUPPORTED contenant *fr*

kevin \$

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

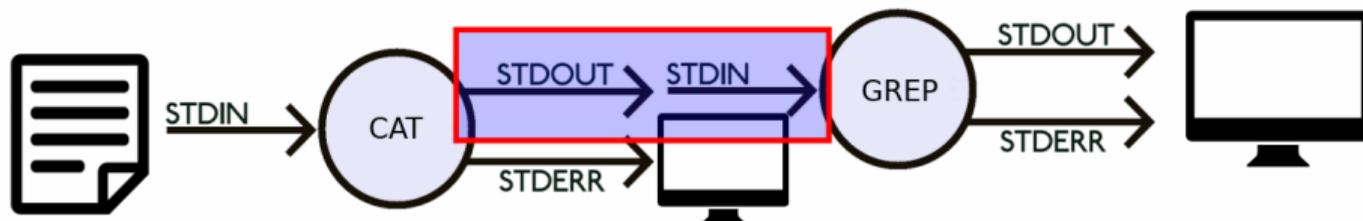
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Figure – Mettre un tuyau entre *cat* et *grep*Affichage des lignes du fichier SUPPORTED contenant *fr*

```
kevin $ cat < /usr/share/i18n/SUPPORTED | grep fr
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les valeurs de retour

Liste les lignes du fichier SUPPORTED contenant *fr*

```
kevin $ cat /usr/share/i18n/SUPPORTED | grep fr
```

```
fr_BE.UTF-8 UTF-8
```

```
fr_BE ISO-8859-1
```

```
fr_BE@euro ISO-8859-15
```

Les valeurs de retour

Liste les lignes du fichier SUPPORTED contenant *fr*

```
kevin $ cat /usr/share/i18n/SUPPORTED | grep fr
```

```
fr_BE.UTF-8 UTF-8
```

```
fr_BE ISO-8859-1
```

```
fr_BE@euro ISO-8859-15
```

Affichage de la valeur de retour de la dernière commande (contenue dans la variable \$?)

```
kevin $ echo $?
```

```
0
```

Liste les lignes du fichier SUPPORTED contenant *fr*

```
kevin $ cat /usr/share/i18n/SUPPORTED | grep fr
```

```
fr_BE.UTF-8 UTF-8
```

```
fr_BE ISO-8859-1
```

```
fr_BE@euro ISO-8859-15
```

Affichage de la valeur de retour de la dernière commande (contenue dans la variable \$?)

```
kevin $ echo $?
```

```
0
```

Liste les lignes du fichier SUPPORTED contenant *not_existing*

```
kevin $ cat /usr/share/i18n/SUPPORTED | grep not_existing
```

Liste les lignes du fichier SUPPORTED contenant *fr*

```
kevin $ cat /usr/share/i18n/SUPPORTED | grep fr
```

```
fr_BE.UTF-8 UTF-8
```

```
fr_BE ISO-8859-1
```

```
fr_BE@euro ISO-8859-15
```

Affichage de la valeur de retour de la dernière commande (contenue dans la variable \$?)

```
kevin $ echo $?
```

```
0
```

Liste les lignes du fichier SUPPORTED contenant *not_existing*

```
kevin $ cat /usr/share/i18n/SUPPORTED | grep not_existing
```

Affichage de la valeur de retour de la dernière commande (contenue dans la variable \$?)

```
kevin $ echo $?
```

```
1
```

Les exécutions conditionnelles

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

L'opérateur &&

Les exécutions conditionnelles

L'opérateur &&

- `cmd && echo ok`
- `echo ok` ne sera exécuté que si `cmd` retourne 0.

L'opérateur ||

Les exécutions conditionnelles

L'opérateur &&

- `cmd && echo ok`
- `echo ok` ne sera exécuté que si `cmd` retourne 0.

L'opérateur ||

- `cmd || echo ko`
- `echo ko` ne sera affiché que si `cmd` retourne quelque chose de différent de 0.

Les exécutions conditionnelles

Un exemple complet

```
Écriture de ok si le fichier SUPPORTED contient fr et de ko si ce n'est pas le cas  
kevin $ cat < /usr/share/i18n/SUPPORTED | grep fr && echo ok ||  
echo ko
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les exécutions conditionnelles

Un exemple complet

```
Écriture de ok si le fichier SUPPORTED contient fr et de ko si ce n'est pas le cas  
kevin $ cat < /usr/share/i18n/SUPPORTED | grep fr && echo ok ||  
echo ko
```

- Dans un premier temps cat est exécuté.

Les exécutions conditionnelles

Un exemple complet

```
Écriture de ok si le fichier SUPPORTED contient fr et de ko si ce n'est pas le cas  
kevin $ cat < /usr/share/i18n/SUPPORTED | grep fr && echo ok ||  
echo ko
```

- Dans un premier temps `cat` est exécuté.
- La sortie de `cat` est transmise à `grep`. Si `grep` trouve au moins une ligne ayant *fr* de présent, elle retournera 0. Sinon, elle retournera 1. Elle affichera également ces lignes à l'écran, car *STDOUT* n'est pas redirigé.

Les exécutions conditionnelles

Un exemple complet

```
Écriture de ok si le fichier SUPPORTED contient fr et de ko si ce n'est pas le cas  
kevin $ cat < /usr/share/i18n/SUPPORTED | grep fr && echo ok ||  
echo ko
```

- Dans un premier temps `cat` est exécuté.
- La sortie de `cat` est transmise à `grep`. Si `grep` trouve au moins une ligne ayant *fr* de présent, elle retournera 0. Sinon, elle retournera 1. Elle affichera également ces lignes à l'écran, car *STDOUT* n'est pas redirigé.
- `&&` « garde » à présent l'exécution de la ligne. Les opérations suivantes ne seront réalisées que si la dernière valeur de retour est un 0. Si c'est le cas, `echo ok` sera exécuté.

Un exemple complet

```
Écriture de ok si le fichier SUPPORTED contient fr et de ko si ce n'est pas le cas  
kevin $ cat < /usr/share/i18n/SUPPORTED | grep fr && echo ok ||  
echo ko
```

- Dans un premier temps `cat` est exécuté.
- La sortie de `cat` est transmise à `grep`. Si `grep` trouve au moins une ligne ayant *fr* de présent, elle retournera 0. Sinon, elle retournera 1. Elle affichera également ces lignes à l'écran, car *STDOUT* n'est pas redirigé.
- `&&` « garde » à présent l'exécution de la ligne. Les opérations suivantes ne seront réalisées que si la dernière valeur de retour est un 0. Si c'est le cas, `echo ok` sera exécuté.
- Un « garde » `||` est à présent rencontré. Il ne permet la suite de l'exécution que si la dernière valeur de retour est autre chose que 0. Note : si nous avons exécuté précédemment `echo ok`, la commande aurait probablement retourné 0.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Le fichier /etc/passwd

Le fichier /etc/passwd

Affichage du manuel du fichier *passwd*

```
kevin $ man 5 passwd
```

NAME

```
passwd - the password file
```

DESCRIPTION

```
/etc/passwd contains one line for each user account, with seven fields delimited by colons (":"). These fields are:
```

- o login name
- o optional encrypted password
- o numerical user ID
- o numerical group ID
- o user name or comment field
- o user home directory
- o optional user command interpreter

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Afficher le fichier

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Afficher le fichier

```
cat /etc/passwd
```

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Afficher le fichier

```
cat /etc/passwd
```

Isoler le champ pertinent

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Afficher le fichier

```
cat /etc/passwd
```

Isoler le champ pertinent

```
cut -d: -f1
```

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Afficher le fichier

```
cat /etc/passwd
```

Isoler le champ pertinent

```
cut -d: -f1
```

Trier les lignes

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Afficher le fichier

```
cat /etc/passwd
```

Isoler le champ pertinent

```
cut -d: -f1
```

Trier les lignes

```
sort
```

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Afficher le fichier

```
cat /etc/passwd
```

Isoler le champ pertinent

```
cut -d: -f1
```

Trier les lignes

```
sort
```

N'afficher que les dix premières lignes

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Afficher le fichier

```
cat /etc/passwd
```

Isoler le champ pertinent

```
cut -d: -f1
```

Trier les lignes

```
sort
```

N'afficher que les dix premières lignes

```
head -n 7
```

Utilisation de l'enchaînement

L'énoncé

À l'aide des commandes `cat`, `cut`, `sort` et `head`, affichez le nom des sept premiers comptes utilisateurs existants sur votre ordinateur triés par ordre alphabétique.

Procédons par étapes

Afficher le fichier

```
cat /etc/passwd
```

Isoler le champ pertinent

```
cut -d: -f1
```

Trier les lignes

```
sort
```

N'afficher que les dix premières lignes

```
head -n 7
```

La commande entière

Affichage des 10 1^{er} comptes utilisateurs par ordre alphabétique

```
kevin $ cat /etc/passwd | cut -d: -f1 | sort | head -n 10
```

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Le shell est l'interface Homme / machine.
- Le shell peut être graphique (*GUI*) ou en ligne de commande (*CLI*).
- Plusieurs techniques permettent de trouver assez efficacement de l'aide sur les logiciels à disposition.
- Les shells *CLI* sont particulièrement puissants : ils peuvent être utilisés comme de véritables langages de programmation.

Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs**
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs
- Les sauvegardes

Conclusion

Exercice 3.5

Les utilisateurs

Création d'un utilisateur

```
root #
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Création d'un utilisateur

```
root # useradd guest
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root #
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root # userdel guest
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root # userdel guest
```

Changement du mot de passe de l'utilisateur courant

```
kevin $
```

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root # userdel guest
```

Changement du mot de passe de l'utilisateur courant

```
kevin $ passwd
```

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root # userdel guest
```

Changement du mot de passe de l'utilisateur courant

```
kevin $ passwd
```

Changement du mot de passe d'un autre utilisateur

```
root #
```

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root # userdel guest
```

Changement du mot de passe de l'utilisateur courant

```
kevin $ passwd
```

Changement du mot de passe d'un autre utilisateur

```
root # passwd guest
```

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root # userdel guest
```

Changement du mot de passe de l'utilisateur courant

```
kevin $ passwd
```

Changement du mot de passe d'un autre utilisateur

```
root # passwd guest
```

Exécution de `ls` via l'utilisateur `root` à l'aide de `sudo`

```
kevin $
```

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root # userdel guest
```

Changement du mot de passe de l'utilisateur courant

```
kevin $ passwd
```

Changement du mot de passe d'un autre utilisateur

```
root # passwd guest
```

Exécution de `ls` via l'utilisateur `root` à l'aide de `sudo`

```
kevin $ sudo ls
```

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root # userdel guest
```

Changement du mot de passe de l'utilisateur courant

```
kevin $ passwd
```

Changement du mot de passe d'un autre utilisateur

```
root # passwd guest
```

Exécution de `ls` via l'utilisateur `root` à l'aide de `sudo`

```
kevin $ sudo ls
```

Changement d'identité de `root` à `guest`

```
root #
```

Création d'un utilisateur

```
root # useradd guest
```

Suppression d'un utilisateur

```
root # userdel guest
```

Changement du mot de passe de l'utilisateur courant

```
kevin $ passwd
```

Changement du mot de passe d'un autre utilisateur

```
root # passwd guest
```

Exécution de `ls` via l'utilisateur `root` à l'aide de `sudo`

```
kevin $ sudo ls
```

Changement d'identité de `root` à `guest`

```
root # su guest
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La propriété des fichiers

Listing détaillé des fichiers dans \$HOME

```
kevin $ ls -l $HOME
```

```
-rwxr-x---. 1 root math 108 24 janv. 20:45 addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Listing détaillé des fichiers dans \$HOME

```
kevin $ ls -l $HOME
```

```
-rwxr-x---. 1 root math 108 24 janv. 20:45 addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Listing détaillé des fichiers dans \$HOME

```
kevin $ ls -l $HOME
```

```
-rwxr-x---. 1 root math 108 24 janv. 20:45 addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La propriété des fichiers

Listing détaillé des fichiers dans \$HOME

```
kevin $ ls -l $HOME
```

```
-rwxr-x---. 1 root math 108 24 janv. 20:45 addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La propriété des fichiers

Listing détaillé des fichiers dans \$HOME

```
kevin $ ls -l $HOME
```

```
-rwxr-x---. 1 root math 108 24 janv. 20:45 addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Listing détaillé des fichiers dans \$HOME

```
kevin $ ls -l $HOME
-rwxr-x---. 1 root math 108 24 janv. 20:45 addition.sh
```

- Un fichier possède un utilisateur et un groupe.
- Trois types de droits « génériques » existent :
 - Le droit de lecture (r).
 - Le droit d'écriture (w).
 - Le droit d'exécution (x).
- Les droits peuvent être donnés à trois catégories d'utilisateurs :
 - L'utilisateur propriétaire.
 - Le groupe propriétaire.
 - Les autres.
- Il existe aussi des droits spéciaux : *setuid*, *setgid* et le *sticky bit*.

Récupération des propriétés depuis d'un fichier avec la commande stat

```
kevin $ stat infal-51.pdf
```

```
Fichier : infal-51.pdf
```

```
Taille : 18411321 Blocs : 35960 Blocs d'E/S : 4096 fichier
```

```
Périphérique : 30h/48d Inœud : 5670978 Liens : 1
```

```
Accès : (0644/-rw-r--r--) UID : ( 1000/kevin) GID : ( 1000/kevin)
```

```
Accès : 2019-01-02 18:36:36.864609708 +0100
```

```
Modif. : 2019-01-02 18:36:36.834609270 +0100
```

```
Changt : 2019-01-02 18:36:36.834609270 +0100
```

```
Créé : -
```

Le bit *setuid*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Le bit *setuid*

- Sur un dossier : chaque nouveau fichier appartiendra à l'utilisateur propriétaire du dossier parent, et non pas à l'auteur de la création.
- Sur un fichier : à l'exécution, le processus prendra l'identité de son propriétaire, et pas celle de l'auteur lançant la commande.
- Exemple : le logiciel `passwd` permettant de mettre à jour son mot de passe.
- Exemple : on assigne le bit *setuid* à un dossier de log dans lequel plusieurs utilisateurs peuvent écrire et lire.

Le bit *setgid*

Introduction

Contexte

Le noyau
UNIX
Linux
Les distributions
Conclusion

Installation

Xubuntu
Les mises à jour
Gestion des logiciels

Les bases

GNU
Le shell
Les utilisateurs
Les fichiers
Les processus
Les scripts

Les données

Le stockage
Btrfs
Les sauvegardes

Conclusion

Le bit *setuid*

- Sur un dossier : chaque nouveau fichier appartiendra à l'utilisateur propriétaire du dossier parent, et non pas à l'auteur de la création.
- Sur un fichier : à l'exécution, le processus prendra l'identité de son propriétaire, et pas celle de l'auteur lançant la commande.
- Exemple : le logiciel `passwd` permettant de mettre à jour son mot de passe.
- Exemple : on assigne le bit *setuid* à un dossier de log dans lequel plusieurs utilisateurs peuvent écrire et lire.

Le bit *setgid*

- Similaire au *setuid* bit mais pour les groupes.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Des droits spéciaux

Le sticky bit

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Le sticky bit

- Sur un fichier : il restera en mémoire même une fois son exécution terminée. Cette utilisation est aujourd'hui obsolète.
- Sur un dossier : les nouveaux fichiers ne seront destructibles que par l'utilisateur propriétaire, même si d'autres peuvent y accéder en écriture.
- Droit utilisé pour le répertoire `/tmp`.

Manipulation des droits

Lègue de la propriété du fichier `addition.sh` à l'utilisateur *kevin* et au groupe *math*

```
kevin $
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Manipulation des droits

Lègue de la propriété du fichier `addition.sh` à l'utilisateur *kevin* et au groupe *math*

```
kevin $ chown kevin:math addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Lègue de la propriété du fichier `addition.sh` à l'utilisateur *kevin* et au groupe *math*

```
kevin $ chown kevin:math addition.sh
```

Lègue de l'ensemble des fichiers dans `/root/script` au groupe *math*

```
kevin $
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Lègue de la propriété du fichier `addition.sh` à l'utilisateur *kevin* et au groupe *math*

```
kevin $ chown kevin:math addition.sh
```

Lègue de l'ensemble des fichiers dans `/root/script` au groupe *math*

```
kevin $ chown -R :math /root/script
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Lègue de la propriété du fichier `addition.sh` à l'utilisateur *kevin* et au groupe *math*

```
kevin $ chown kevin:math addition.sh
```

Lègue de l'ensemble des fichiers dans `/root/script` au groupe *math*

```
kevin $ chown -R :math /root/script
```

Modification des droits en 750 pour le fichier `addition.sh` (version octale)

```
kevin $
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Lègue de la propriété du fichier `addition.sh` à l'utilisateur *kevin* et au groupe *math*

```
kevin $ chown kevin:math addition.sh
```

Lègue de l'ensemble des fichiers dans `/root/script` au groupe *math*

```
kevin $ chown -R :math /root/script
```

Modification des droits en 750 pour le fichier `addition.sh` (version octale)

```
kevin $ chmod 750 addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Lègue de la propriété du fichier `addition.sh` à l'utilisateur *kevin* et au groupe *math*

```
kevin $ chown kevin:math addition.sh
```

Lègue de l'ensemble des fichiers dans `/root/script` au groupe *math*

```
kevin $ chown -R :math /root/script
```

Modification des droits en 750 pour le fichier `addition.sh` (version octale)

```
kevin $ chmod 750 addition.sh
```

Modification des droits en 750 pour le fichier `addition.sh` (version symbolique)

```
kevin $
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Lègue de la propriété du fichier `addition.sh` à l'utilisateur *kevin* et au groupe *math*

```
kevin $ chown kevin:math addition.sh
```

Lègue de l'ensemble des fichiers dans `/root/script` au groupe *math*

```
kevin $ chown -R :math /root/script
```

Modification des droits en 750 pour le fichier `addition.sh` (version octale)

```
kevin $ chmod 750 addition.sh
```

Modification des droits en 750 pour le fichier `addition.sh` (version symbolique)

```
kevin $ chmod u=rwx,g=rx,o= addition.sh
```

Le groupe primaire

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Le groupe primaire

- Un utilisateur ne peut être que dans un seul groupe primaire.
- C'est celui qui sera par défaut assigné à un fichier créé par l'utilisateur en question.
- C'est aussi celui qui sera assigné aux processus que l'utilisateur lancera.

Les groupes secondaires

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Le groupe primaire

- Un utilisateur ne peut être que dans un seul groupe primaire.
- C'est celui qui sera par défaut assigné à un fichier créé par l'utilisateur en question.
- C'est aussi celui qui sera assigné aux processus que l'utilisateur lancera.

Les groupes secondaires

- Autres groupes auxquels un utilisateur a accès.
- Permettent principalement le partage de fichiers.
- Exemple : les utilisateurs du groupe *audio* peuvent accéder à la carte son.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 3.6

L'arborescence

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les chemins absolus

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les chemins absolus

- Position d'un fichier depuis la racine du système de fichier root.
- Indépendant du répertoire de travail du processus.
- Commence toujours par un slash.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les chemins absolus

- Position d'un fichier depuis la racine du système de fichier root.
- Indépendant du répertoire de travail du processus.
- Commence toujours par un slash.

Les chemins relatifs

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les chemins absolus

- Position d'un fichier depuis la racine du système de fichier root.
- Indépendant du répertoire de travail du processus.
- Commence toujours par un slash.

Les chemins relatifs

- Position d'un fichier depuis le répertoire de travail du processus courant.
- Ne commencent jamais par un slash.

Les chemins absolus

- Position d'un fichier depuis la racine du système de fichier root.
- Indépendant du répertoire de travail du processus.
- Commence toujours par un slash.

Les chemins relatifs

- Position d'un fichier depuis le répertoire de travail du processus courant.
- Ne commencent jamais par un slash.

Exemple

L'instance courante de *Bash* a comme répertoire de travail `/home/kevin/script`.

La commande `: bash calculette.sh` serait équivalente à `bash`

`/home/kevin/script/calculette.sh`, la première version appelle le script a interpréter depuis un chemin relatif, la seconde depuis son équivalent absolu.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les commandes utiles

- Retourne le chemin absolu correspondant au répertoire de travail du processus courant

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les commandes utiles

- Retourne le chemin absolu correspondant au répertoire de travail du processus courant
 - `pwd`

Les commandes utiles

- Retourne le chemin absolu correspondant au répertoire de travail du processus courant
 - `pwd`
- Permet de changer le répertoire de travail du processus courant.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Retourne le chemin absolu correspondant au répertoire de travail du processus courant
 - `pwd`
- Permet de changer le répertoire de travail du processus courant.
 - `cd $HOME`
 - `cd ..`
 - `cd /usr/share/doc`

Les commandes utiles

- Retourne le chemin absolu correspondant au répertoire de travail du processus courant
 - `pwd`
- Permet de changer le répertoire de travail du processus courant.
 - `cd $HOME`
 - `cd ..`
 - `cd /usr/share/doc`
- Liste le contenu d'un répertoire.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Retourne le chemin absolu correspondant au répertoire de travail du processus courant
 - `pwd`
- Permet de changer le répertoire de travail du processus courant.
 - `cd $HOME`
 - `cd ..`
 - `cd /usr/share/doc`
- Liste le contenu d'un répertoire.
 - `ls .` ou simplement `ls`
 - `ls script`
 - `ls /home/kevin/script`

Liste du contenu d'un répertoire

```
kevin $ ls -l
total 8
drwxr-xr-x 3  kevin  kevin 120 24 déc. 12:27 .
drwxrwxrwt 19  root   root  440 24 déc. 12:24 ..
-rw-r--r-- 1  kevin  kevin  90 24 déc. 12:24 file.txt
drwxr-xr-x 2  kevin  kevin  40 24 déc. 12:24 folder
lrwxrwxrwx 1  kevin  kevin   8 24 déc. 12:24 link.txt -> file.txt
```

Liste du contenu d'un répertoire

```
kevin $ ls -l
total 8
drwxr-xr-x 3 kevin kevin 120 24 déc. 12:27 .
drwxrwxrwt 19 root root 440 24 déc. 12:24 ..
-rw-r--r-- 1 kevin kevin 90 24 déc. 12:24 file.txt
drwxr-xr-x 2 kevin kevin 40 24 déc. 12:24 folder
lrwxrwxrwx 1 kevin kevin 8 24 déc. 12:24 link.txt -> file.txt
```

Type de fichier

- La lettre b, c ou - représentent un fichier (*bloc, caractère* ou normal).
- La lettre d représente un dossier.
- La lettre l représente un lien symbolique.
- Voir `info ls` pour la liste complète.

Liste du contenu d'un répertoire

```
kevin $ ls -l
total 8
drwxr-xr-x 3 kevin kevin 120 24 déc. 12:27 .
drwxrwxrwt 19 root root 440 24 déc. 12:24 ..
-rw-r--r-- 1 kevin kevin 90 24 déc. 12:24 file.txt
drwxr-xr-x 2 kevin kevin 40 24 déc. 12:24 folder
lrwxrwxrwx 1 kevin kevin 8 24 déc. 12:24 link.txt -> file.txt
```

Droits du fichier

- Chaque fichier peut avoir des droits de lecture (r), d'écriture (w) ou d'exécution (x);
- Les droits peuvent être appliqués pour l'utilisateur, le groupe et les autres.
- Il existe des droits spéciaux, cf : `info ls`.

Liste du contenu d'un répertoire

```
kevin $ ls -l
total 8
drwxr-xr-x 3 kevin kevin 120 24 déc. 12:27 .
drwxrwxrwt 19 root root 440 24 déc. 12:24 ..
-rw-r--r-- 1 kevin kevin 90 24 déc. 12:24 file.txt
drwxr-xr-x 2 kevin kevin 40 24 déc. 12:24 folder
lrwxrwxrwx 1 kevin kevin 8 24 déc. 12:24 link.txt -> file.txt
```

Nombre de liens durs sur l'inode sous-jacente

- Une fois ce nombre à zéro, le fichier est considéré comme supprimé.
- Un dossier parent a au moins autant de liens durs que de dossiers enfants.

Liste du contenu d'un répertoire

```
kevin $ ls -l
total 8
drwxr-xr-x 3 kevin kevin 120 24 déc. 12:27 .
drwxrwxrwt 19 root root 440 24 déc. 12:24 ..
-rw-r--r-- 1 kevin kevin 90 24 déc. 12:24 file.txt
drwxr-xr-x 2 kevin kevin 40 24 déc. 12:24 folder
lrwxrwxrwx 1 kevin kevin 8 24 déc. 12:24 link.txt -> file.txt
```

Propriétaires du fichier

- La première colonne représente l'utilisateur ;
- Et la seconde son groupe.

Liste du contenu d'un répertoire

```
kevin $ ls -l
total 8
drwxr-xr-x 3 kevin kevin 120 24 déc. 12:27 .
drwxrwxrwt 19 root root 440 24 déc. 12:24 ..
-rw-r--r-- 1 kevin kevin 90 24 déc. 12:24 file.txt
drwxr-xr-x 2 kevin kevin 40 24 déc. 12:24 folder
lrwxrwxrwx 1 kevin kevin 8 24 déc. 12:24 link.txt -> file.txt
```

Taille du fichier

- Par défaut la taille est en octets.
- L'option `-h` à la commande permet de mettre la taille à une échelle plus lisible.

Liste du contenu d'un répertoire

```
kevin $ ls -l
total 8
drwxr-xr-x 3  kevin  kevin 120 24 déc. 12:27 .
drwxrwxrwt 19 root   root 440 24 déc. 12:24 ..
-rw-r--r-- 1  kevin  kevin 90  24 déc. 12:24 file.txt
drwxr-xr-x 2  kevin  kevin 40  24 déc. 12:24 folder
lrwxrwxrwx 1  kevin  kevin 8   24 déc. 12:24 link.txt -> file.txt
```

Dernière date de modification du fichier

Liste du contenu d'un répertoire

```
kevin $ ls -l
total 8
drwxr-xr-x 3 kevin kevin 120 24 déc. 12:27 .
drwxrwxrwt 19 root root 440 24 déc. 12:24 ..
-rw-r--r-- 1 kevin kevin 90 24 déc. 12:24 file.txt
drwxr-xr-x 2 kevin kevin 40 24 déc. 12:24 folder
lrwxrwxrwx 1 kevin kevin 8 24 déc. 12:24 link.txt -> file.txt
```

Nom du fichier

Dans le cas du lien symbolique, nous avons en plus sa référence d'affichée.

Liste du contenu d'un répertoire

```
kevin $ ls -l
total 8
drwxr-xr-x 3  kevin  kevin 120 24 déc. 12:27 .
drwxrwxrwt 19  root   root  440 24 déc. 12:24 ..
-rw-r--r-- 1  kevin  kevin  90 24 déc. 12:24 file.txt
drwxr-xr-x 2  kevin  kevin  40 24 déc. 12:24 folder
lrwxrwxrwx 1  kevin  kevin   8 24 déc. 12:24 link.txt -> file.txt
```

Taille du dossier actuel

- La taille est exprimées en blocs alloués.
- Sauf avec l'option `-h`. En mode « human readable » la taille est définie en octets.
- Le calcul n'est pas récursif : les tailles des dossiers enfants ne sont pas incluses.

Filesystem Hierarchy Standard

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Filesystem Hierarchy Standard

- Permet de structurer l'emplacement des fichiers dans l'arborescence du système d'exploitation.
- Lisible depuis la page de manuel *hier* : `man hier`.
- N'est strictement respectée nulle part...
- Des exemples :

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Filesystem Hierarchy Standard

- Permet de structurer l'emplacement des fichiers dans l'arborescence du système d'exploitation.
- Lisible depuis la page de manuel *hier* : `man hier`.
- N'est strictement respectée nulle part...
- Des exemples :
 - `/etc` Fichiers de configuration.
 - `/home` Dossiers personnels des utilisateurs du système.
 - `/lib/modules` Module du noyau chargeable.
 - `/tmp` Fichiers temporaires.
 - `/usr/bin` Logiciels utilisables (dossier principal).
 - `/usr/share/man` Localisation des pages de manuels.
 - `/usr/src/linux` Emplacement des fichiers sources de Linux (si installés).

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 3.7

Les types de fichiers

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les commandes utiles

Les commandes utiles

- Création d'un fichier :

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les commandes utiles

- Création d'un fichier :
 - `touch mémoires.txt`
 - `touch /home/kevin/mémoires.txt`
 - `nano mémoires.txt`
- Création d'un répertoire :

Les commandes utiles

- Création d'un fichier :
 - `touch mémoires.txt`
 - `touch /home/kevin/mémoires.txt`
 - `nano mémoires.txt`
- Création d'un répertoire :
 - `mkdir documents`
 - `mkdir /home/kevin/documents`
- Supprimer des fichiers :

Les commandes utiles

- Création d'un fichier :
 - `touch mémoires.txt`
 - `touch /home/kevin/mémoires.txt`
 - `nano mémoires.txt`
- Création d'un répertoire :
 - `mkdir documents`
 - `mkdir /home/kevin/documents`
- Supprimer des fichiers :
 - `rm mémoires.txt`
 - `rmdir documents`
 - `rm -r documents`
- Déplacer des fichiers :

- Création d'un fichier :
 - `touch mémoires.txt`
 - `touch /home/kevin/mémoires.txt`
 - `nano mémoires.txt`
- Création d'un répertoire :
 - `mkdir documents`
 - `mkdir /home/kevin/documents`
- Supprimer des fichiers :
 - `rm mémoires.txt`
 - `rmdir documents`
 - `rm -r documents`
- Déplacer des fichiers :
 - `cp mémoires.txt documents`
 - `cp /home/kevin/mémoires.txt documents`
 - `cp -R documents /mnt/usb1`

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

« Tout est fichier »

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

« Tout est fichier »

- Les disques durs sont des fichiers.
- Les cartes son sont des fichiers.
- Les interfaces réseau sont des fichiers.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

« Tout est fichier »

- Les disques durs sont des fichiers.
- Les cartes son sont des fichiers.
- Les interfaces réseau sont des fichiers.

→ Le fichier est une interface entre le matériel et l'espace utilisateur qui est gérée par le noyau à l'aide de drivers.

Les dossiers */proc*, */dev* et */sys*

/dev

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les dossiers */proc*, */dev* et */sys*

/dev

- Interface dans l'espace utilisateur permettant de manipuler les périphériques via le noyau et les drivers.
- Géré aujourd'hui par *udev* sur pratiquement toutes les distributions.

/proc

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les dossiers */proc*, */dev* et */sys**/dev*

- Interface dans l'espace utilisateur permettant de manipuler les périphériques via le noyau et les drivers.
- Géré aujourd'hui par *udev* sur pratiquement toutes les distributions.

/proc

- Interface de manipulation des processus.
- Il existe un répertoire */proc/<pid>* pour chaque programme en cours d'exécution.
- `man procfs`

Les dossiers */proc*, */dev* et */sys*

/sys

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les dossiers */proc*, */dev* et */sys**/sys*

- Interface de manipulation du noyau.
- `man sysfs`

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les fichiers spéciaux

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les fichiers spéciaux

Les fichiers de type caractères

Les fichiers spéciaux

Les fichiers de type caractères

- Ne peuvent être montés.
- Traitent les entrées/sorties caractère par caractère.
- Ne peuvent être utilisés que par un processus à la fois.
- Exemple : l'interface de communication avec le clavier, la carte son ou le réseau.

Les fichiers de type blocs

Les fichiers de type caractères

- Ne peuvent être montés.
- Traitent les entrées/sorties caractère par caractère.
- Ne peuvent être utilisés que par un processus à la fois.
- Exemple : l'interface de communication avec le clavier, la carte son ou le réseau.

Les fichiers de type blocs

- Peuvent être montés.
- Utilisé principalement pour l'écriture de fichiers.
- Transmission des informations bloc par bloc.
- Gestion asynchrone.
- Exemple : l'accès aux disques durs, clef usb, etc.

Listing détaillé des fichiers dans /dev

```
kevin $ ls /dev -l
```

```
brw-rw---- 1 root disk 8, 16 25 janv. 07:59 sdb  
brw-rw---- 1 root disk 8, 17 25 janv. 07:59 sdb1  
brw-rw---- 1 root disk 8, 18 25 janv. 07:59 sdb2  
crw--w---- 1 root tty 4, 0 25 janv. 07:59 tty0  
crw--w---- 1 root tty 4, 1 25 janv. 07:59 tty1
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Listing détaillé des fichiers dans /dev

```
kevin $ ls /dev -l
brw-rw---- 1 root disk 8, 16 25 janv. 07:59 sdb
brw-rw---- 1 root disk 8, 17 25 janv. 07:59 sdb1
brw-rw---- 1 root disk 8, 18 25 janv. 07:59 sdb2
crw--w---- 1 root  tty 4,  0 25 janv. 07:59 tty0
crw--w---- 1 root  tty 4,  1 25 janv. 07:59 tty1
```

- Le premier caractère de la commande `ls -l` désigne le type de fichier. La lettre `b`, `c` ou `-` représentent un fichier (respectivement de type *bloc*, *caractère* ou *normal*).

Listing détaillé des fichiers dans /dev

```
kevin $ ls /dev -l
brw-rw---- 1 root disk 8, 16 25 janv. 07:59 sdb
brw-rw---- 1 root disk 8, 17 25 janv. 07:59 sdb1
brw-rw---- 1 root disk 8, 18 25 janv. 07:59 sdb2
crw--w---- 1 root tty 4, 0 25 janv. 07:59 tty0
crw--w---- 1 root tty 4, 1 25 janv. 07:59 tty1
```

- Le premier caractère de la commande `ls -l` désigne le type de fichier. La lettre `b`, `c` ou `-` représentent un fichier (respectivement de type *bloc*, *caractère* ou *normal*).
- Ces fichiers sont gérés par des drivers identifiés par leur numéro majeur ainsi qu'un paramètre lui étant transmis : le numéro mineur.

Listing détaillé des fichiers dans /dev

```
kevin $ ls /dev -l  
brw-rw---- 1 root disk 8, 16 25 janv. 07:59 sdb  
brw-rw---- 1 root disk 8, 17 25 janv. 07:59 sdb1  
brw-rw---- 1 root disk 8, 18 25 janv. 07:59 sdb2  
crw--w---- 1 root tty 4, 0 25 janv. 07:59 tty0  
crw--w---- 1 root tty 4, 1 25 janv. 07:59 tty1
```

- Le premier caractère de la commande `ls -l` désigne le type de fichier. La lettre `b`, `c` ou `-` représentent un fichier (respectivement de type *bloc*, *caractère* ou *normal*).
- Ces fichiers sont gérés par des drivers identifiés par leur numéro majeur ainsi qu'un paramètre lui étant transmis : le numéro mineur.
- Les fichiers « normaux » ne sont en fait que des entités abstraites gérées par des drivers utilisant *VFS*.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Gérer les fichiers spéciaux

udev

udev

- Sur les systèmes récents, le logiciel *udev* se charge de créer automatiquement les fichiers de `/dev` en fonction du matériel listé par le noyau.
- *udev* peut être configuré à l'aide de règles.
- `udevadm info /dev/random` permet d'avoir des informations sur la façon dont *udev* a créé le fichier `/dev/random`.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

udev

- Sur les systèmes récents, le logiciel *udev* se charge de créer automatiquement les fichiers de `/dev` en fonction du matériel listé par le noyau.
- *udev* peut être configuré à l'aide de règles.
- `udevadm info /dev/random` permet d'avoir des informations sur la façon dont *udev* a créé le fichier `/dev/random`.

mknod

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau
UNIX
Linux
Les distributions
Conclusion

Installation

Xubuntu
Les mises à jour
Gestion des logiciels

Les bases

GNU
Le shell
Les utilisateurs
Les fichiers
Les processus
Les scripts

Les données

Le stockage
Btrfs
Les sauvegardes

Conclusion

udev

- Sur les systèmes récents, le logiciel *udev* se charge de créer automatiquement les fichiers de `/dev` en fonction du matériel listé par le noyau.
- *udev* peut être configuré à l'aide de règles.
- `udevadm info /dev/random` permet d'avoir des informations sur la façon dont *udev* a créé le fichier `/dev/random`.

mknod

- `mknod /tmp/random c 1 8`
- Création d'un fichier spécial de type caractère à l'emplacement `/tmp/random` ayant le numéro de driver majeur 1 et mineur 8. C'est l'équivalent de `/dev/random`.

Les inodes (ou inœuds, en français)

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les inodes (ou inœuds, en français)

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Contient les « métadonnées » d'un fichier.
- La norme *POSIX* décrit une grande partie des attributs.
- La commande `stat` permet de voir l'inode d'un fichier.
- Quelques exemples d'informations :
 - Dernière date de modification.
 - Dernière date de lecture.
 - Ses propriétaires.
 - Ses permissions.
 - Le nombre de blocs alloués au fichier (sa taille).
 - Son emplacement sur le disque.
- D'autres infos peuvent être présentes en fonction du système de fichier.

Les liens durs

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Ils permettent d'avoir un seul fichier à plusieurs endroits.

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l  
-rw-r--r-- 1 root root 108 24 janv. 20:45 addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Ils permettent d'avoir un seul fichier à plusieurs endroits.

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l  
-rw-r--r-- 1 root root 108 24 janv. 20:45 addition.sh
```

Création d'un lien dur entre le fichier addition.sh et addition_second.sh

```
kevin $ ln addition.sh addition_second.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Ils permettent d'avoir un seul fichier à plusieurs endroits.

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l
-rw-r--r-- 1 root root 108 24 janv. 20:45 addition.sh
```

Création d'un lien dur entre le fichier addition.sh et addition_second.sh

```
kevin $ ln addition.sh addition_second.sh
```

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l
-rw-r--r-- 2 root root 108 24 janv. 20:45 addition_second.sh
-rw-r--r-- 2 root root 108 24 janv. 20:45 addition.sh
```

Ils permettent d'avoir un seul fichier à plusieurs endroits.

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l
-rw-r--r-- 1 root root 108 24 janv. 20:45 addition.sh
```

Création d'un lien dur entre le fichier addition.sh et addition_second.sh

```
kevin $ ln addition.sh addition_second.sh
```

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l
-rw-r--r-- 2 root root 108 24 janv. 20:45 addition_second.sh
-rw-r--r-- 2 root root 108 24 janv. 20:45 addition.sh
```

Exemple : les fichiers . et .. présents dans tous les répertoires.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les liens symboliques

Ils permettent d'avoir une redirection d'un emplacement vers un second.

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l
```

```
-rw-r--r-- 1 root root 108 24 janv. 20:45 addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Ils permettent d'avoir une redirection d'un emplacement vers un second.

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l
```

```
-rw-r--r-- 1 root root 108 24 janv. 20:45 addition.sh
```

Création d'un lien symbolique entre addition.sh et addition_second.sh

```
kevin $ ln -s addition.sh addition_second.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Ils permettent d'avoir une redirection d'un emplacement vers un second.

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l
-rw-r--r-- 1 root root 108 24 janv. 20:45 addition.sh
```

Création d'un lien symbolique entre addition.sh et addition_second.sh

```
kevin $ ln -s addition.sh addition_second.sh
```

Listing des fichiers dans le répertoire courant

```
kevin $ ls -l
lrwxrwxrwx 1 root root 11 27 janv. 15:35 addition_second.sh ->
addition.sh
-rw-r--r-- 1 root root 108 24 janv. 20:45 addition.sh
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 3.8

Les processus

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La commande : ps

La commande : ps

Affichage d'un instantané des processus courants

```
kevin $ ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	842	1	0	janv.27	?	00:00:00	/usr/sbin/sshd -D
root	12276	9157	0	12:39	pts/1	00:00:00	ps -ef

→ Commande ayant énormément d'options différentes.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La commande : top

La commande : top

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

```
top - 12:49:57 up 15:22, 1 user, load average: 0,21, 0,06, 0,02
Tasks: 217 total, 2 running, 176 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,3 us, 0,3 sy, 0,0 ni, 99,0 id, 0,0 wa, 0,3 hi, 0,0 si, 0,0 st
KiB Mem : 2044224 total, 103712 free, 1095516 used, 844996 buff/cache
KiB Swap: 1261564 total, 1261564 free, 0 used. 717692 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1618	khagner	20	0	3176304	330504	97992	S	1,0	16,2	2:14.39	gnome-shell
447	root	20	0	0	0	0	S	0,3	0,0	0:00.32	jbd2/dm-0-8
1299	gdm	20	0	779860	23628	18428	S	0,3	1,2	0:09.62	gsd-color
12610	khagner	20	0	162296	4736	4028	R	0,3	0,2	0:00.05	top
1	root	20	0	164912	11880	7612	S	0,0	0,6	0:04.55	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.03	kthreadd
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La commande : htop

Petit retour sur /proc

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Petit retour sur /proc

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

```
[root@myrphak khagner]# ps axjf | grep man
12549 14042 14042 12549 pts/0 14042 S+ 1000 0:00 | | \_ man procfs
14042 14053 14042 12549 pts/0 14042 S+ 1000 0:00 | | \_ man procfs
13229 14252 14251 13170 pts/1 14251 S+ 0 0:00 | | \_ grep --color=auto man
[root@myrphak khagner]# cat /proc/12549/cmdline
bash[root@myrphak khagner]# cat /proc/14042/cmdline
manprocfs[root@myrphak khagner]# cat /proc/14042/stat
14042 (man) S 12549 14042 12549 34816 14042 4210688 3786 2326 0 0 1 2 16 2 20 0 1 0 5681952 129536000
891 18446744073709551615 94415784099840 94415784197560 140735506294128 0 0 0 4102 81921 1 0 0 17 0 0
0 0 0 94415786297216 94415786301972 94415795167232 140735506301679 140735506301690 140735506301690
140735506305003 0
```

Petit retour sur /proc

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

```
[root@myrphak khagner]# ps axjf | grep man
12549 14042 14042 12549 pts/0    14042 S+   1000   0:00 |   |   \_ man procfs
14042 14053 14042 12549 pts/0    14042 S+   1000   0:00 |   |   \_ man procfs
13229 14252 14251 13170 pts/1    14251 S+     0   0:00 |   |   \_ grep --color=auto man
[root@myrphak khagner]# cat /proc/12549/cmdline
bash[root@myrphak khagner]# cat /proc/14042/cmdline
manprocfs[root@myrphak khagner]# cat /proc/14042/stat
14042 (man) S 12549 14042 12549 34816 14042 4210688 3786 2326 0 0 1 2 16 2 20 0 1 0 5681952 129536000
891 18446744073709551615 94415784099840 94415784197560 140735506294128 0 0 0 4102 81921 1 0 0 17 0 0
0 0 0 94415786297216 94415786301972 94415795167232 140735506301679 140735506301690 140735506301690
140735506305003 0

/proc/[pid]/stat
      Status information about the process.  This is used by ps(1).  It is defined in the
      kernel source file fs/proc/array.c.
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

```
[root@myrphak khagner]# ps axjf | grep man
12549 14042 14042 12549 pts/0    14042 S+   1000    0:00 |   |   \_ man procfs
14042 14053 14042 12549 pts/0    14042 S+   1000    0:00 |   |   \_ man procfs
13229 14252 14251 13170 pts/1    14251 S+     0    0:00 |   |   \_ grep --color=auto man
[root@myrphak khagner]# cat /proc/12549/cmdline
bash[root@myrphak khagner]# cat /proc/14042/cmdline
manprocfs[root@myrphak khagner]# cat /proc/14042/stat
14042 (man) S 12549 14042 12549 34816 14042 4210688 3786 2326 0 0 1 2 16 2 20 0 1 0 5681952 129536000
891 18446744073709551615 94415784099840 94415784197560 140735506294128 0 0 0 4102 81921 1 0 0 17 0 0
0 0 0 94415786297216 94415786301972 94415795167232 140735506301679 140735506301690 140735506301690
140735506305003 0

/proc/[pid]/stat
      Status information about the process.  This is used by ps(1).  It is defined in the
      kernel source file fs/proc/array.c.
```

Le logiciel `ps` utilise `/proc` (`procfs`) pour récupérer les informations sur les autres processus en cours d'exécution. → Tout est fichier !

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les signaux

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- `man 7 signal`.
- Permet à un processus d'envoyer un « message » à un autre.
 - Lui demander de s'arrêter (SIGINT).
 - Demander au noyau d'arrêter le processus (SIGKILL).
 - Demander au noyau de stopper son exécution (SIGSTOP).
 - Lui demander de reprendre son exécution (SIGCONT).
 - Etc...
- SIGSTOP et SIGKILL sont toujours gérés par le noyau.
- Le comportement des autres signaux peut être défini au niveau de l'application destinatrice.
- $\text{^C (contrôle + C) = SIGINT}$.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Tuer un processus

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Tuer un processus

La commande kill

Tuer un processus

La commande kill

- `kill -s INT <pid>` ou `kill -2 <pid>`
- `kill -s KILL <pid>` ou `kill -9 <pid>`

SIGINT vs SIGKILL

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La commande kill

- `kill -s INT <pid>` ou `kill -2 <pid>`
- `kill -s KILL <pid>` ou `kill -9 <pid>`

SIGINT vs SIGKILL

- SIGINT peut être réécrit par l'application. C'est donc un moyen sûr de demander l'arrêt d'un logiciel, car celui-ci peut se faire correctement.
- SIGKILL par contre est géré par le noyau qui va directement mettre un terme au processus, sans lui permettre de stopper son exécution correctement.
- SIGINT est donc à privilégier, mais SIGKILL peut s'avérer nécessaire si le processus a crashé.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Copie d'un processus.
- Unique moyen de créer un nouveau processus sous Linux.
- Les deux processus sont ensuite complètement isolés au niveau de la mémoire (si l'un crash, l'autre continue de fonctionner).
- Possède un nouveau *PID*.
- Son *PPID* est le *PID* du processus effectuant le fork.
- Exemple : un fork a lieu à chaque nouvelle exécution d'une commande dans *Bash*.

Les processus zombies

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Copie d'un processus.
- Unique moyen de créer un nouveau processus sous Linux.
- Les deux processus sont ensuite complètement isolés au niveau de la mémoire (si l'un crash, l'autre continue de fonctionner).
- Possède un nouveau *PID*.
- Son *PPID* est le *PID* du processus effectuant le fork.
- Exemple : un fork a lieu à chaque nouvelle exécution d'une commande dans *Bash*.

Les processus zombies

Le processus parent doit toujours récupérer la valeur de retour de ses enfants (via le syscall *wait*). Si un parent se termine avant la récupération de la valeur de retour de son enfant, il restera bloqué, car le processus n'est réellement supprimé du noyau qu'au moment où sa valeur de retour est récupérée par quelqu'un. Un processus zombie a donc été créé : il ne sera plus jamais exécuté, mais il demeurera en mémoire.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- *Light weight process*
- La mémoire est partagée entre le processus parent et ses threads.
- Si le processus se termine, tous ses threads sont tués.
- Plus dangereux, car la mémoire étant partagée, il peut y avoir des accès concurrents.
- Partage le même *PID* que son parent, mais a un autre *SPID*.
- Exemple : un navigateur web peut posséder un thread actualisant l'affichage d'une page en fonction de son défilement, un autre pour décoder une vidéo en streaming pendant qu'un troisième télécharge un fichier.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- *Light weight process*
 - La mémoire est partagée entre le processus parent et ses threads.
 - Si le processus se termine, tous ses threads sont tués.
 - Plus dangereux, car la mémoire étant partagée, il peut y avoir des accès concurrents.
 - Partage le même *PID* que son parent, mais a un autre *SPID*.
 - Exemple : un navigateur web peut posséder un thread actualisant l'affichage d'une page en fonction de son défilement, un autre pour décoder une vidéo en streaming pendant qu'un troisième télécharge un fichier.
- Le fork permet de faciliter l'instantanéité au sein d'une même application.

L'arborescence des processus

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

L'arborescence des processus

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- 0 idle : le processus est exécuté quand aucun autre n'a besoin de l'être.
- 1 init : ce processus charge tous les autres logiciels. C'est également lui qui s'occupe des arrêts et redémarrages. Aujourd'hui, il s'agit habituellement de *systemd*.
- 2 kthreadd : processus internes au noyau.
- 3 - ∞ : tous les autres processus sont des fils plus ou moins directs de 1 ou 2.

L'arborescence des processus

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

0 idle : le processus est exécuté quand aucun autre n'a besoin de l'être.

1 init : ce processus charge tous les autres logiciels. C'est également lui qui s'occupe des arrêts et redémarrages. Aujourd'hui, il s'agit habituellement de *systemd*.

2 kthreadd : processus internes au noyau.

3 - ∞ : tous les autres processus sont des fils plus ou moins directs de 1 ou 2.

Visualisation de tous les processus avec l'arborescence

```
kevin $ pstree -Taph 0
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La suppression du processus parent

Les processus orphelins

La suppression du processus parent

Les processus orphelins

Quand un processus parent est supprimé, tous ses enfants encore en cours d'exécution se voient rattachés au *PID* 1 par le kernel.

Le cas du shell

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

La suppression du processus parent

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les processus orphelins

Quand un processus parent est supprimé, tous ses enfants encore en cours d'exécution se voient rattachés au *PID* 1 par le kernel.

Le cas du shell

Beaucoup de shells (dont *Bash*) envoient le signal *SIGHUP* à tous ses processus enfants quand l'utilisateur quitte le shell. En général, *SIGHUP* a pour conséquence l'arrêt du programme.

Le builtin *Bash* `disown` peut être utilisé pour explicitement demander au shell de ne pas envoyer *SIGHUP* lors de sa fermeture.

PID Identifiant du processus.

NI Valeur *nice*. Indique la priorité du processus (-19 est la plus forte, 20 la plus faible).

%CPU $\text{cputime} / \text{realtime}$: si le processus existe depuis une seconde et que pendant celle-ci, il occupait un processeur pendant 25 centièmes, son pourcentage d'utilisation sera de $0.25 / 1 = 25\%$.

%MEM pourcentage de mémoire physique totale utilisé par le processus sur la mémoire totale disponible.

TIME temps cumulé d'utilisation du *CPU* pour le processus.

Load average représente la charge du système. Ça correspond au nombre de processus en cours d'utilisation et ceux en attente de la libération du *CPU* ou d'entrées/sorties.

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Des informations et de l'aide sont disponibles directement via les pages de manuels et le dossier `/usr/share/doc`.
- Il n'y a pas qu'une façon de faire quelque chose.
- Les shells permettent d'utiliser de façon efficace les outils du noyau, comme l'exécution de nouveaux logiciels.
- Uniformité de l'interface : tout est fichier.
- Les droits sont affinés à l'aide de la notion d'utilisateurs.
- Les processus permettent de partager le temps d'allocation des *CPU*.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 3.9

Les scripts

Un script minimaliste

Fichier *bonjour.sh*

Fichier *bonjour.sh*

```
#!/usr/bin/bash  
echo "Bonjour";  
exit 3;
```

Un script minimaliste

Exécution

Exécution du fichier `bonjour.sh`

```
kevin $ bash bonjour.sh
```

```
Bonjour
```

Affichage de la valeur de retour

```
kevin $ echo $?
```

```
3
```

Fichier *bonjour.sh*

```
#!/usr/bin/bash  
echo "Bonjour";  
exit 3;
```

Le shebang

Indique à l'OS que le fichier n'est pas un binaire *ELF* mais un script qui devra être exécuté via l'interpréteur indiqué (ici */usr/bin/bash*).

Un script minimaliste

Exécution

```
Exécution du fichier bonjour.sh  
kevin $ bash bonjour.sh  
Bonjour
```

```
Affichage de la valeur de retour  
kevin $ echo $?  
3
```

Fichier *bonjour.sh*

```
#!/usr/bin/bash  
echo "Bonjour";  
exit 3;
```

Le shebang

Indique à l'OS que le fichier n'est pas un binaire *ELF* mais un script qui devra être exécuté via l'interpréteur indiqué (ici `/usr/bin/bash`).

La valeur de retour

Nous pouvons voir que l'instruction `exit` permet de définir la valeur de la variable de retour que notre script aura.

Un script minimaliste

Exécution

```
Exécution du fichier bonjour.sh  
kevin $ bash bonjour.sh  
Bonjour
```

```
Affichage de la valeur de retour  
kevin $ echo $?  
3
```

Fichier *addition.sh*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les variables

Fichier *addition.sh*

```
#!/usr/bin/bash
```

```
VARIABLE_1=4;
```

```
VARIABLE_2=5;
```

```
VARIABLE_3=$((
```

```
    $VARIABLE_1 + $VARIABLE_2
```

```
));
```

```
echo $VARIABLE_3;
```

Exécution du fichier *addition.sh*

```
kevin $ bash addition.sh
```

```
9
```

Les variables

Fichier *addition.sh*

```
#!/usr/bin/bash
```

```
VARIABLE_1=4;
```

```
VARIABLE_2=5;
```

```
VARIABLE_3=$((
```

```
    $VARIABLE_1 + $VARIABLE_2
```

```
));
```

```
echo $VARIABLE_3;
```

Déclaration de variables Le signe égal doit toucher le nom de la variable et la valeur. On ne préfixe pas le nom de la variable par un dollar.

```
Exécution du fichier addition.sh
kevin $ bash addition.sh
9
```

Les variables

Fichier *addition.sh*

```
#!/usr/bin/bash
```

```
VARIABLE_1=4;
```

```
VARIABLE_2=5;
```

```
VARIABLE_3=$((
```

```
    $VARIABLE_1 + $VARIABLE_2
```

```
));
```

```
echo $VARIABLE_3;
```

Déclaration de variables Le signe égal doit toucher le nom de la variable et la valeur. On ne préfixe pas le nom de la variable par un dollar.

Lecture de variables On doit préfixer la variable avec un dollar.

```
Exécution du fichier addition.sh
kevin $ bash addition.sh
9
```

Fichier *addition.sh*

```
#!/usr/bin/bash
```

```
VARIABLE_1=4;
```

```
VARIABLE_2=5;
```

```
VARIABLE_3=$((
```

```
    $VARIABLE_1 + $VARIABLE_2
```

```
));
```

```
echo $VARIABLE_3;
```

Déclaration de variables Le signe égal doit toucher le nom de la variable et la valeur. On ne préfixe pas le nom de la variable par un dollar.

Lecture de variables On doit préfixer la variable avec un dollar.

Réalisation de calculs Les calculs en *Bash* doivent être fait entre les instructions `=$((et))`.

```

Exécution du fichier addition.sh
kevin $ bash addition.sh
9

```

Fichier *calculette.sh*

Les paramètres

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Fichier *calculette.sh*

```
#!/usr/bin/bash
```

```
RESULTAT=$((  
    $1 + $2  
));  
echo $RESULTAT;
```

Les paramètres

Exécution de *calculette.sh*

```
kevin $ bash calculette.sh 2 3  
5
```

Exécution avec un seul paramètre

```
kevin $ bash calculette.sh 2  
calculette.sh: ligne 5:  
2 + : erreur de syntaxe : opérande  
attendu (le symbole erroné est  
« + »)
```

Fichier *calculette.sh*

```
#!/usr/bin/bash
```

```
RESULTAT=$((
```

```
    $1 + $2
```

```
));
```

```
echo $RESULTAT;
```

Les paramètres de l'application

Les variables \$1 à \$9 permettent de récupérer les 9 premiers paramètres du script.

Les paramètres

Exécution de *calculette.sh*

```
kevin $ bash calculette.sh 2 3
5
```

Exécution avec un seul paramètre

```
kevin $ bash calculette.sh 2
calculette.sh: ligne 5:
2 + : erreur de syntaxe : opérande
attendu (le symbole erroné est
« + »)
```

Les paramètres

Fichier *calculette.sh*

```
#!/usr/bin/bash
```

```
RESULTAT=$((
    $1 + $2
));
echo $RESULTAT;
```

Les paramètres de l'application

Les variables \$1 à \$9 permettent de récupérer les 9 premiers paramètres du script.

Attention

Ce code n'est pas très robuste, il peut facilement planter. Il nous faudrait vérifier les paramètres avant de les utiliser !

Exécution de calculette.sh

```
kevin $ bash calculette.sh 2 3
5
```

Exécution avec un seul paramètre

```
kevin $ bash calculette.sh 2
calculette.sh: ligne 5:
2 + : erreur de syntaxe : opérande
attendu (le symbole erroné est
« + »)
```

Les conditions

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Fichier *devinette.sh*

```
#!/bin/bash
```

```
echo "De ... beau ?"  
read REPONSE;
```

```
if test "$REPONSE" = "bleu"; then  
    echo "Gagné !";  
else  
    echo "Perdu.";   
fi
```

Les conditions

Exécution de *devinette.sh*

```
kevin $ bash devinette.sh  
De ... beau ?  
bleu  
Gagné !
```

Exécution de *devinette.sh*

```
kevin $ bash devinette.sh  
De ... beau ?  
vert  
Perdu.
```

Les conditions

Fichier *devinette.sh*

```
#!/bin/bash
```

```
echo "De ... beau ?"
```

```
read REPONSE;
```

```
if test "$REPONSE" = "bleu"; then
```

```
    echo "Gagné !";
```

```
else
```

```
    echo "Perdu.";
```

```
fi
```

Exécution de *devinette.sh*

```
kevin $ bash devinette.sh
```

```
De ... beau ?
```

```
bleu
```

```
Gagné !
```

Exécution de *devinette.sh*

```
kevin $ bash devinette.sh
```

```
De ... beau ?
```

```
vert
```

```
Perdu.
```

Fichier *devinette.sh*

```
#!/bin/bash
```

```
echo "De ... beau ?"  
read REPONSE;
```

```
if test "$REPONSE" = "bleu"; then  
    echo "Gagné !";  
else  
    echo "Perdu.";   
fi
```

Les conditions

Exécution de *devinette.sh*

```
kevin $ bash devinette.sh  
De ... beau ?  
bleu  
Gagné !
```

Exécution de *devinette.sh*

```
kevin $ bash devinette.sh  
De ... beau ?  
vert  
Perdu.
```

Les conditions

Fichier *devinette.sh*

```
#!/bin/bash
```

```
echo "De ... beau ?"  
read REPONSE;
```

```
if test "$REPONSE" = "bleu"; then  
    echo "Gagné !";  
else  
    echo "Perdu.";   
fi
```

Exécution de *devinette.sh*

```
kevin $ bash devinette.sh  
De ... beau ?  
bleu  
Gagné !
```

Exécution de *devinette.sh*

```
kevin $ bash devinette.sh  
De ... beau ?  
vert  
Perdu.
```

Fichier *bombe.sh*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

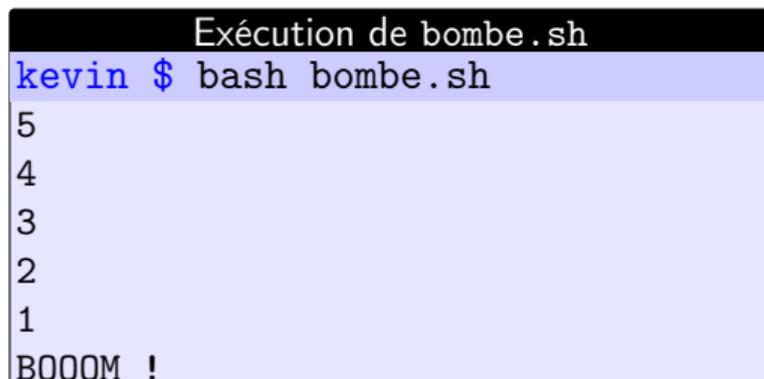
Les sauvegardes

Conclusion

Fichier *bombe.sh*

```
#!/usr/bin/bash

TEMPS_RESTANT=5;
while test $TEMPS_RESTANT -ne 0
do
    echo $TEMPS_RESTANT;
    sleep 1;
    TEMPS_RESTANT=$((
        $TEMPS_RESTANT - 1
    ));
done
echo "BOOOM !";
```

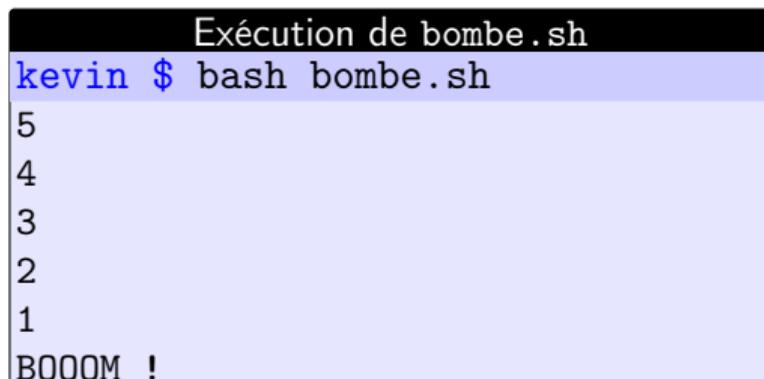


```
Exécution de bombe.sh
kevin $ bash bombe.sh
5
4
3
2
1
BOOOM !
```

Fichier *bombe.sh*

```
#!/usr/bin/bash

TEMPS_RESTANT=5;
while test $TEMPS_RESTANT -ne 0
do
    echo $TEMPS_RESTANT;
    sleep 1;
    TEMPS_RESTANT=$((
        $TEMPS_RESTANT - 1
    ));
done
echo "BOOOM !";
```



```
Exécution de bombe.sh
kevin $ bash bombe.sh
5
4
3
2
1
BOOOM !
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 4.1

Le disque dur

Les disques durs *HDD*

Figure – Un disque dur (*crédit*[🔗])



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les disques *solid-state SSD*

Figure – Un disque solid-state (*crédit* )



HDD vs SSD

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Des technologies différentes.
- Les *HDD* font tourner physiquement des disques sous des têtes de lecture pour récupérer les données.
- Les *SSD* utilisent de la mémoire flash.
- Les façons de fonctionner sont totalement différentes entre les deux types de disques, pourtant, ils doivent avoir un comportement uniforme pour l'utilisateur.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les partitions

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Permettent de virtuellement découper un disque.
- Utilisées par exemple pour séparer les données du système d'exploitation.
- Utilisées aussi pour pouvoir installer deux systèmes d'exploitation en parallèle.
- Permet d'utiliser plusieurs systèmes de fichiers.
- Plusieurs formats de table de partitions existent, dont *MBR* et *GPT*.

Le système de partitionnement *MBR* (ou *MsDos*)

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Le système de partitionnement *MBR* (ou *MsDos*)

- Système le plus répandu sur les anciennes installations, date des années 80.
- *MBR = Master Boot Record*
- Le *MBR* est un espace réservé au début du disque contenant :
 - Le programme de boot sur les 446 premiers bits.
 - La signature du disque.
 - La table des partitions, permettant d'avoir jusqu'à quatre partitions primaires.
 - La valeur *55AA* qui indique la fin du *MBR*.
- Taille max du disque limitée à 2 Tio.
- Utilise un adressage Cylindre/Tête/Secteur (*CHS*).

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Le système de partitionnement *MBR* (ou *MsDos*)

- Système le plus répandu sur les anciennes installations, date des années 80.
- *MBR = Master Boot Record*
- Le *MBR* est un espace réservé au début du disque contenant :
 - Le programme de boot sur les 446 premiers bits.
 - La signature du disque.
 - La table des partitions, permettant d'avoir jusqu'à quatre partitions primaires.
 - La valeur *55AA* qui indique la fin du *MBR*.
- Taille max du disque limitée à 2 Tio.
- Utilise un adressage *Sylindre/Tête/Secteur (CHS)*.

Structure of a classical generic MBR

Address		Description	Size (bytes)
Hex	Dec		
+000 _{hex}	+0	Bootstrap code area	446
+1BE _{hex}	+446	Partition entry №1	Partition table (for primary partitions)
+1CE _{hex}	+462	Partition entry №2	
+1DE _{hex}	+478	Partition entry №3	
+1EE _{hex}	+494	Partition entry №4	
+1FE _{hex}	+510	55 _{hex}	Boot signature ^[a]
+1FF _{hex}	+511	AA _{hex}	
Total size: 446 + 4×16 + 2			512

Le système de partitionnement *GPT*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

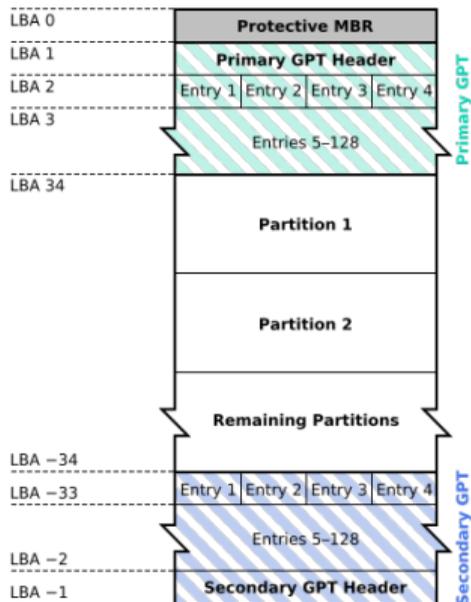
Btrfs

Les sauvegardes

Conclusion

Le système de partitionnement *GPT*

GUID Partition Table Scheme



- *GUID Partition Table*.
- La *GPT* est un sous-ensemble de spécifications de *UEFI*.
- Non compatible avec les *BIOS*.
- N'ont plus les limitations des tables *MBR*.
- L'*UEFI* est capable de directement booter sur une partition.
- Utilise un adressage par bloc (*LBA*).

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Affichage des partitions *SCSI* du système

```
kevin $ ls -l /dev | grep sd  
brw-rw---- 1 root disk 8, 0 25 janv. 07:59 sda  
brw-rw---- 1 root disk 8, 1 25 janv. 07:59 sda1  
brw-rw---- 1 root disk 8, 2 25 janv. 07:59 sda2  
brw-rw---- 1 root disk 8, 16 25 janv. 07:59 sdb  
brw-rw---- 1 root disk 8, 17 25 janv. 07:59 sdb1  
brw-rw---- 1 root disk 8, 18 25 janv. 07:59 sdb2
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Affichage des partitions *SCSI* du système

```
kevin $ ls -l /dev | grep sd  
brw-rw---- 1 root disk 8, 0 25 janv. 07:59 sda  
brw-rw---- 1 root disk 8, 1 25 janv. 07:59 sda1  
brw-rw---- 1 root disk 8, 2 25 janv. 07:59 sda2  
brw-rw---- 1 root disk 8, 16 25 janv. 07:59 sdb  
brw-rw---- 1 root disk 8, 17 25 janv. 07:59 sdb1  
brw-rw---- 1 root disk 8, 18 25 janv. 07:59 sdb2
```

→ Le noyau, avec l'aide des différents drivers permet de nous abstraire toutes ces différences techniques.

Exercice 4.2

Les systèmes de fichiers

Différences entre le système de fichier et la partition

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Différences entre le système de fichier et la partition

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Le système de fichier représente le format avec lequel une partition sera structurée.
- Un système de fichier peut venir d'ailleurs, comme du réseau (*NFS*, *sshfs*), d'un fichier (un cdrom) ou être généré depuis un programme (*procf*s).
- Un système de fichier possède un point de montage dans l'arborescence « principale » formant ainsi une arborescence unique composée en réalité de plusieurs systèmes de fichiers venant de supports physiques ou virtuels différents.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Les systèmes *ext*

ext2

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

ext2

Format historique de *Linux* qui améliore son système de fichier initial *ext* qui était repris de *Minix*.

ext3

Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage**
- Btrfs
- Les sauvegardes

Conclusion

ext2

Format historique de *Linux* qui améliore son système de fichier initial *ext* qui était repris de *Minix*.

ext3

Similaire à *ext2*, avec la journalisation en plus. Cette évolution lui permet de devenir très stable, car en cas de crash de la machine ou de coupure d'électricité, le journal permet de réparer les données. Les corruptions sont presque devenues inexistantes.

ext4

Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs
- Les sauvegardes

Conclusion

ext2

Format historique de *Linux* qui améliore son système de fichier initial *ext* qui était repris de *Minix*.

ext3

Similaire à *ext2*, avec la journalisation en plus. Cette évolution lui permet de devenir très stable, car en cas de crash de la machine ou de coupure d'électricité, le journal permet de réparer les données. Les corruptions sont presque devenues inexistantes.

ext4

Ajoute des améliorations de performances, notamment au niveau de la gestion des fichiers avec de longs noms et au niveau de la fragmentation. C'est le système par défaut utilisé actuellement par beaucoup de distribution *Linux*.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Non *POSIX*!
- Ne pourrait être utilisé comme système de fichier sous *Linux* car il ne supporte pas la gestion des droits.
- Nous pouvons néanmoins monter une partition *NTFS* et y lire et ajouter des fichiers.

A Case for Redundant Arrays of Inexpensive Disks

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

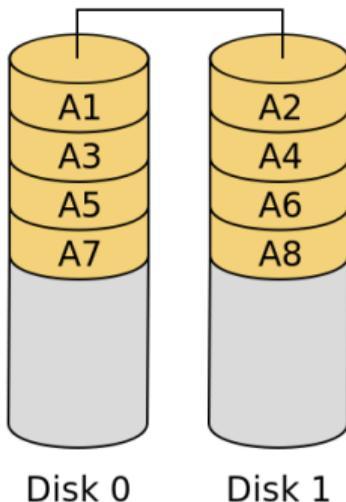
Btrfs

Les sauvegardes

Conclusion

A Case for Redundant Arrays of Inexpensive Disks

RAID 0



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

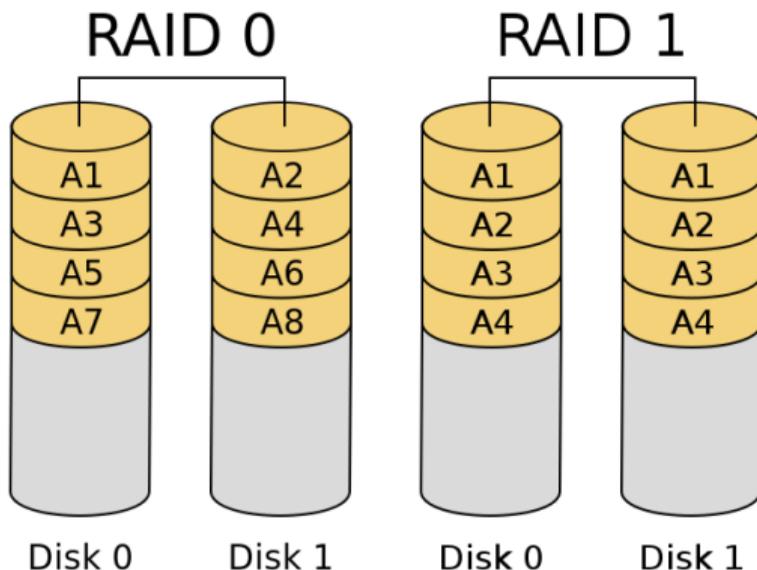
Le stockage

Btrfs

Les sauvegardes

Conclusion

A Case for Redundant Arrays of Inexpensive Disks



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

A Case for Redundant Arrays of Inexpensive Disks

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

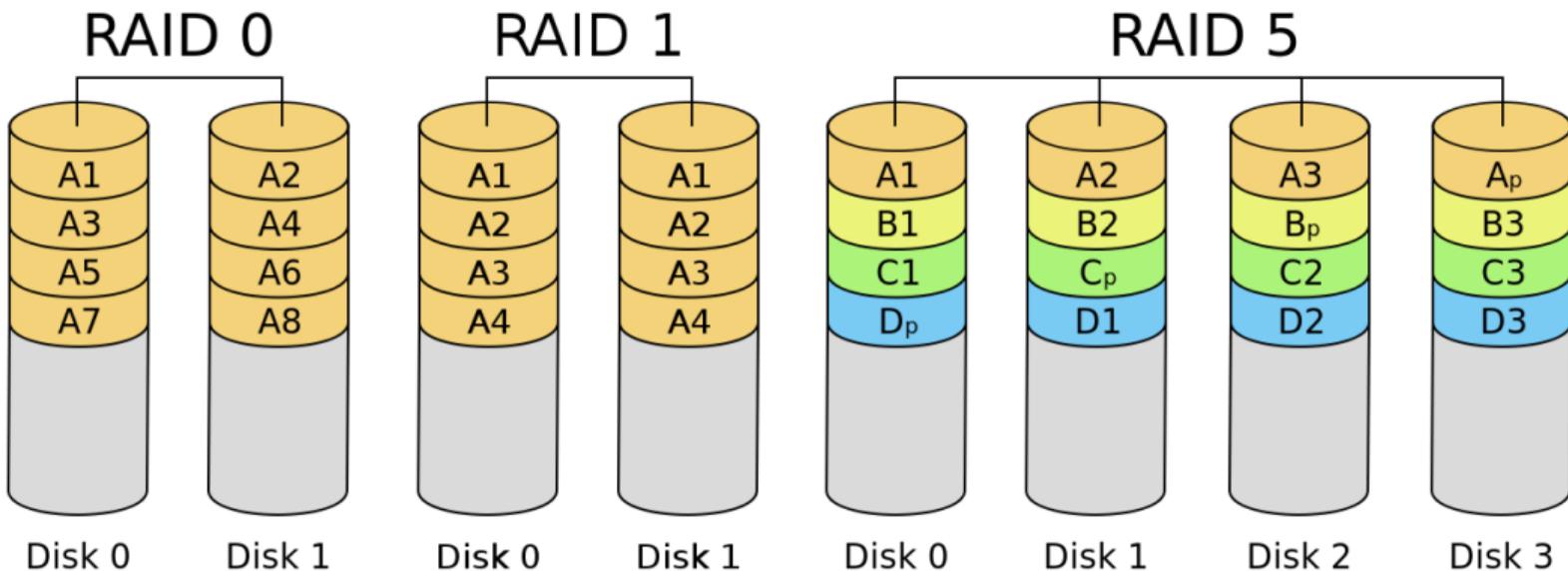
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Futur successeur probable à *ext4* qui apporte des fonctionnalités très intéressantes :

- *RAID* logiciel.
- Instantanés (snapshots).
- Compression automatique.
- Redimensionnement à chaud.
- Déduplication.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

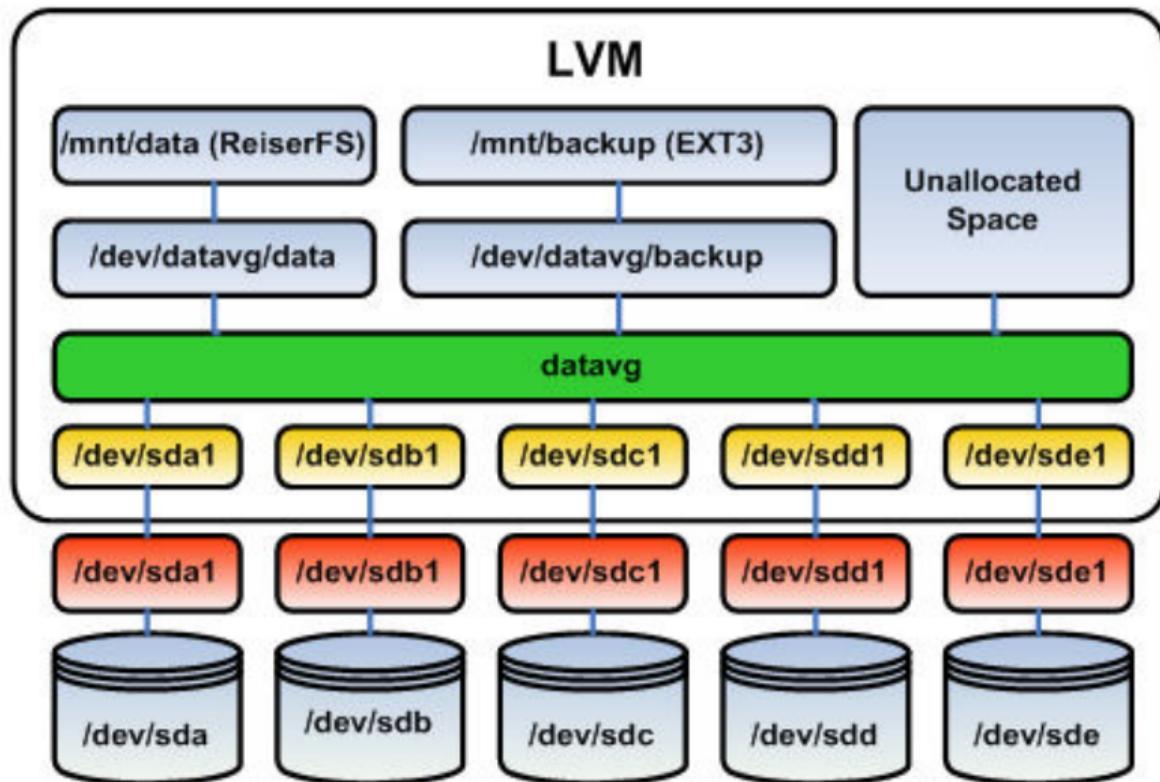
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs**
- Les sauvegardes

Conclusion

Exercice 4.3

Travail pratique

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

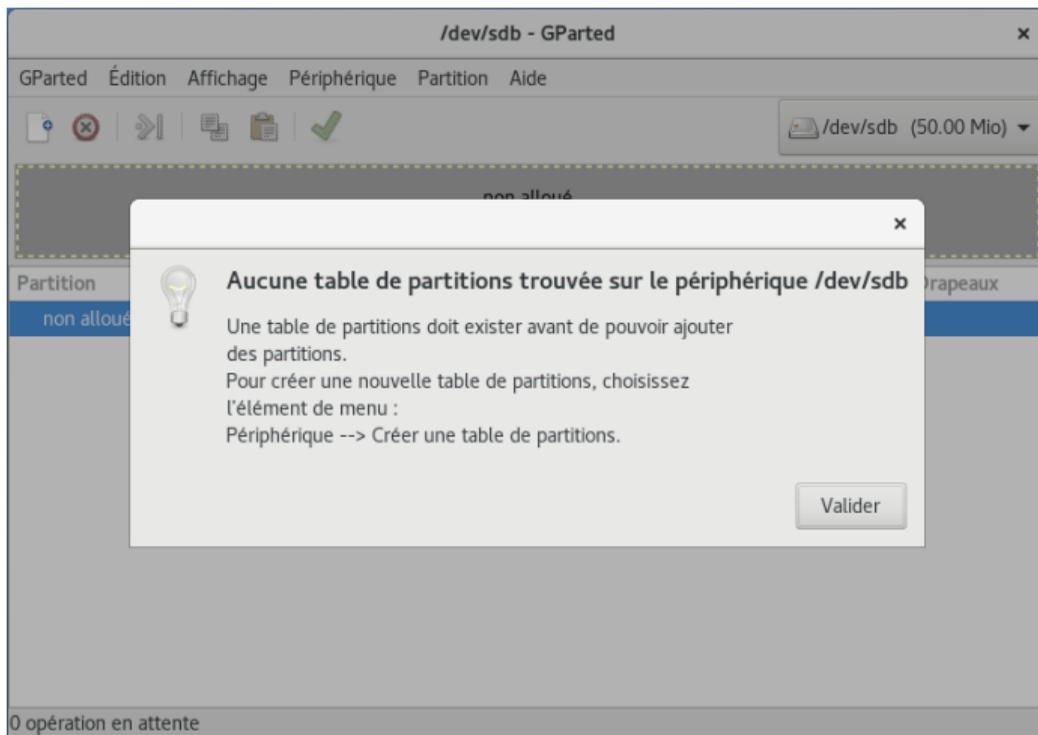
Conclusion

Comment voir les nouveaux disques ?

Comment voir les nouveaux disques ?

Affichage des partitions *SCSI* du système

```
root # ls -l /dev | grep sd  
brw-rw---- 1 root disk 8, 0 31 janv. 07:49 sda  
brw-rw---- 1 root disk 8, 1 31 janv. 07:49 sda1  
brw-rw---- 1 root disk 8, 2 31 janv. 07:49 sda2  
brw-rw---- 1 root disk 8, 16 31 janv. 07:49 sdb  
brw-rw---- 1 root disk 8, 32 31 janv. 07:49 sdc  
brw-rw---- 1 root disk 8, 48 31 janv. 07:49 sdd
```

Formatage d'une partie de *sdb*

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Formatage d'une partie de *sdb*

/dev/sdb - GParted

GParted Édition Affichage Périphérique Partition Aide

/dev/sdb (50.00 Mio)

Nouvelle partition #1
30.00 Mio

non alloué
19.00 Mio

Partition	Système de fichiers	Taille	Utilisé	Inutilisé	Drapeaux
Nouvelle partition #1	ext4	30.00 Mio	---	---	
non alloué	non alloué	19.00 Mio	---	---	

Supprimer /dev/sdb1 (ext4, 20.00 Mio) de /dev/sdb
 Créer Partition primaire #1 (ext4, 30.00 Mio) sur /dev/sdb

2 opérations en attente

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Voir la nouvelle partition

Voir la nouvelle partition

Affichage des partitions *SCSI* du système

```
root # ls -l /dev | grep sd  
brw-rw---- 1 root disk 8, 0 31 janv. 08:06 sda  
brw-rw---- 1 root disk 8, 1 31 janv. 08:06 sda1  
brw-rw---- 1 root disk 8, 2 31 janv. 08:06 sda2  
brw-rw---- 1 root disk 8, 16 31 janv. 08:06 sdb  
brw-rw---- 1 root disk 8, 17 31 janv. 08:06 sdb1  
brw-rw---- 1 root disk 8, 32 31 janv. 08:06 sdc  
brw-rw---- 1 root disk 8, 48 31 janv. 08:06 sdd
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Monter le système de fichier

Monter le système de fichier

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Création du dossier /mnt/mon_nouveau_disque

```
root # mkdir /mnt/mon_nouveau_disque
```

Montage de /dev/sdb1 sur /mnt/mon_nouveau_disque

```
root # mount /dev/sdb1 /mnt/mon_nouveau_disque
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Vérifier l'espace disque libre

Vérifier l'espace disque libre

Affichage de l'espace disque libre sur tous les points de montage

kevin \$ df -h

Sys. de fichiers	Taille	Utilisé	Dispo	Uti%	Monté sur
devtmpfs	986M	0	986M	0%	/dev
tmpfs	999M	0	999M	0%	/dev/shm
tmpfs	999M	1,7M	997M	1%	/run
tmpfs	999M	0	999M	0%	/sys/fs/cgroup
/dev/mapper/...	9,6G	6,9G	2,3G	76%	/
tmpfs	999M	156K	999M	1%	/tmp
/dev/sda1	976M	175M	735M	20%	/boot
partage	237G	172G	65G	73%	/media/sf_partage
tmpfs	200M	5,2M	195M	3%	/run/user/1000
/dev/sr0	56M	56M	0	100%	/run/media/kevin/VBox_GAs
/dev/sdb1	29M	371K	26M	1%	/mnt/mon_nouveau_disque

Vérifier les inodes libres

Affichage des inœuds restants sur les différents système de fichiers

kevin \$ df -ih

Sys. de fichiers	Inœuds	IUtil.	ILibre	IUtil%	Monté sur
devtmpfs	247K	484	246K	1%	/dev
tmpfs	250K	1	250K	1%	/dev/shm
tmpfs	250K	852	249K	1%	/run
tmpfs	250K	17	250K	1%	/sys/fs/cgroup
/dev/mapper/...	628K	206K	422K	33%	/
tmpfs	250K	28	250K	1%	/tmp
/dev/sda1	64K	452	64K	1%	/boot
partage	1000	0	1000	0%	/media/sf_partage
tmpfs	250K	49	250K	1%	/run/user/1000
/dev/sr0	0	0	0	-	/run/media/kevin/VBox_GAs
/dev/sdb1	7,1K	11	7,1K	1%	/mnt/mon_nouveau_disque

Vérifier les inodes libres

Affichage des inœuds restants sur les différents système de fichiers

kevin \$ df -ih

Sys. de fichiers	Inœuds	IUtil.	ILibre	IUtil%	Monté sur
devtmpfs	247K	484	246K	1%	/dev
tmpfs	250K	1	250K	1%	/dev/shm
tmpfs	250K	852	249K	1%	/run
tmpfs	250K	17	250K	1%	/sys/fs/cgroup
/dev/mapper/...	628K	206K	422K	33%	/
tmpfs	250K	28	250K	1%	/tmp
/dev/sda1	64K	452	64K	1%	/boot
partage	1000	0	1000	0%	/media/sf_partage
tmpfs	250K	49	250K	1%	/run/user/1000
/dev/sr0	0	0	0	-	/run/media/kevin/VBox_GAs
/dev/sdb1	7,1K	11	7,1K	1%	/mnt/mon_nouveau_disque

Téléchargement et déplacement de la vidéo

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Téléchargement de la vidéo sur notre bureau

```
kevin $
```

Téléchargement et déplacement de la vidéo

Téléchargement de la vidéo sur notre bureau

```
kevin $ wget https://share.spyzone.fr/cesi/video.mp4 -O  
$USER/Bureau/video.mp4
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Téléchargement et déplacement de la vidéo

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Téléchargement de la vidéo sur notre bureau

```
kevin $ wget https://share.spyzone.fr/cesi/video.mp4 -O  
$USER/Bureau/video.mp4
```

Copie de la vidéo sur le nouveau disque

```
kevin $
```

Téléchargement et déplacement de la vidéo

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Téléchargement de la vidéo sur notre bureau

```
kevin $ wget https://share.spyzone.fr/cesi/video.mp4 -O  
$USER/Bureau/video.mp4
```

Copie de la vidéo sur le nouveau disque

```
kevin $ cp Bureau/video.mp4 /mnt/mon_nouveau_disque/
```

Téléchargement et déplacement de la vidéo

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Téléchargement de la vidéo sur notre bureau

```
kevin $ wget https://share.spyzone.fr/cesi/video.mp4 -O  
$USER/Bureau/video.mp4
```

Copie de la vidéo sur le nouveau disque

```
kevin $ cp Bureau/video.mp4 /mnt/mon_nouveau_disque/  
cp: erreur d'écriture dans '/mnt/mon_nouveau_disque/video.mp4': Aucun  
espace disponible sur le périphérique
```

Création d'un système *RAID 1* avec *btrfs*Création d'un *RAID 1* avec *btrfs*

```
root # mkfs.btrfs -d raid1 /dev/sdb2 /dev/sdc
```

```
UUID: e925907e-1193-458a-bdd4-2f6a780f3c96
```

```
Node size: 16384
```

```
Sector size: 4096
```

```
Filesystem size: 299.00MiB
```

```
Block group profiles:
```

```
    Data: RAID1 64.00MiB
```

```
    Metadata: RAID1 32.00MiB
```

```
    System: RAID1 8.00MiB
```

```
SSD detected: no
```

```
Incompat features: extref, skinny-metadata
```

```
Number of devices: 2
```

```
Devices:
```

ID	SIZE	PATH
1	149.00MiB	/dev/sdb2
2	150.00MiB	/dev/sdc

Montage du système *Btrfs*

Création d'un dossier `/mnt/mon_second_disque`

```
root # mkdir /mnt/mon_second_disque
```

Montage du fichier bloc `/dev/sdc` dans le dossier `/mnt/mon_second_disque`

```
root # mount /dev/sdc /mnt/mon_second_disque
```

Affichage de l'espace dispo et utilisé du *FS* monté sur `/mnt/mon_second_disque`

```
root # df -h | grep mon_second_disque
```

```
/dev/sdb2          205M    17M  158M  10% /mnt/mon_second_disque
```

Déplacement du fichier de travail dans le dossier `/mnt/mon_second_disque`

```
root # cd /mnt/mon_second_disque
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Démontage du FS /mnt/mon_second_disque

```
root # umount /mnt/mon_second_disque
```

Montage du disque /dev/sdc avec la compression d'activée

```
root # mount -o compress=zlib /dev/sdc /mnt/mon_second_disque
```

Listing détaillé des fichiers dans le répertoire /mnt/mon_second_disque

```
root # ls -lh /mnt/mon_second_disque
total 88M -rw-r--r--. 1 root root 88M 5 janv. 16:16
filesystem-defragmentation-strategies.mp4
```

Affichage d'informations sur le FS du fichier /mnt/mon_second_disque

```
root # btrfs filesystem df /mnt/mon_second_disque
Data, RAID1: total=108.00MiB, used=82.00MiB
System, RAID1: total=8.00MiB, used=16.00KiB
Metadata, RAID1: total=32.00MiB, used=208.00KiB
GlobalReserve, single: total=16.00MiB, used=0.00B
```

Démontage du FS /mnt/mon_second_disque

```
root # umount /mnt/mon_second_disque
```

Montage du disque /dev/sdc avec la compression d'activée

```
root # mount -o compress=zlib /dev/sdc /mnt/mon_second_disque
```

Listing détaillé des fichiers dans le répertoire /mnt/mon_second_disque

```
root # ls -lh /mnt/mon_second_disque
total 88M -rw-r--r--. 1 root root 88M 5 janv. 16:16
filesystem-defragmentation-strategies.mp4
```

Affichage d'informations sur le FS du fichier /mnt/mon_second_disque

```
root # btrfs filesystem df /mnt/mon_second_disque
Data, RAID1: total=108.00MiB, used=82.00MiB
System, RAID1: total=8.00MiB, used=16.00KiB
Metadata, RAID1: total=32.00MiB, used=208.00KiB
GlobalReserve, single: total=16.00MiB, used=0.00B
```

Affichage d'informations sur le *FS* du fichier `/mnt/mon_second_disque`

```
root # btrfs filesystem df .
```

```
Data, RAID1: total=108.00MiB, used=82.00MiB[...]
```

Listing des fichiers dans le répertoire `/mnt/mon_second_disque`

```
root # ls -lh /mnt/mon_second_disque
```

```
-rw-r-r-. 1 root root 88M 30 janv. 13:57 filesystem-defragmentation-...
```

Suppression de la vidéo *filesystem-defragmentation-...*

```
root # rm /mnt/mon_second_disque/filesystem-defragmentation-...
```

Nouvelle copie de la vidéo sur le système de fichier

```
root # cp /filesystem-defragm... /mnt/mon_second_disque
```

Affichage d'informations sur le *FS* du fichier `/mnt/mon_second_disque`

```
root # btrfs filesystem df .
```

```
Data, RAID1: total=108.00MiB, used=73.91MiB System, RAID1:  
total=8.00MiB, used=16.00KiB Metadata, RAID1: total=32.00MiB,  
used=144.00KiB[...]
```

Affichage d'informations sur le *FS* du fichier /mnt/mon_second_disque

```
root # btrfs filesystem df .
```

```
Data, RAID1: total=108.00MiB, used=82.00MiB[...]
```

Listing des fichiers dans le répertoire /mnt/mon_second_disque

```
root # ls -lh /mnt/mon_second_disque
```

```
-rw-r-r-. 1 root root 88M 30 janv. 13:57 filesystem-defragmentation-...
```

Suppression de la vidéo *filesystem-defragmentation-...*

```
root # rm /mnt/mon_second_disque/filesystem-defragmentation-...
```

Nouvelle copie de la vidéo sur le système de fichier

```
root # cp /filesystem-defragm... /mnt/mon_second_disque
```

Affichage d'informations sur le *FS* du fichier /mnt/mon_second_disque

```
root # btrfs filesystem df .
```

```
Data, RAID1: total=108.00MiB, used=73.91MiB System, RAID1:  
total=8.00MiB, used=16.00KiB Metadata, RAID1: total=32.00MiB,  
used=144.00KiB[...]
```

Réparer un disque défectueux

Tentative de montage d'un *FS* qui échoue

```
root # mount /dev/sdb2 /mnt/mon_second_disque  
mount: /mnt/mon_second_disque : wrong fs type, bad option, bad  
superblock on /dev/sdb2, missing codepage or helper program,  
or other error.
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Réparer un disque défectueux

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Tentative de montage d'un *FS* qui échoue

```
root # mount /dev/sdb2 /mnt/mon_second_disque
```

```
mount: /mnt/mon_second_disque : wrong fs type, bad option, bad  
superblock on /dev/sdb2, missing codepage or helper program,  
or other error.
```

Formatage de la partition /dev/sdc en *btrfs*

```
root # mkfs.btrfs /dev/sdc
```

Réparer un disque défectueux

Affichage de l'état actuel des systèmes de fichier *btrfs*

```
root # btrfs filesystem show
warning, device 2 is missing
Label: none uuid: b8e2b172-0e91-4854-a611-f1c98ad99638
  Total devices 2 FS bytes used 128.00KiB
devid 1 size 149.00MiB used 104.00MiB path /dev/sdb2
*** Some devices missing
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Réparer un disque défectueux

Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs**
- Les sauvegardes

Conclusion

Montage du *FS* /dev/sdb2 en mode dégradé

```
root # mount -o degraded /dev/sdb2 /mnt/mon_second_disque
```

Listing détaillé des fichiers dans le répertoire /mnt/mon_second_disque

```
root # ls -lh /mnt/mon_second_disque
```

```
total 88M
```

```
-rw-r-r-. 1 root root 88M 30 janv. 13:57 filesystem-defragmentation-...
```

Remplacement du disque 2 du *FS* par la partition /mnt/mon_second_disque

```
root # btrfs replace start -f 2 /dev/sdc /mnt/mon_second_disque/
```

Réparer un disque défectueux

Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs**
- Les sauvegardes

Conclusion

Montage du *FS* /dev/sdb2 en mode dégradé

```
root # mount -o degraded /dev/sdb2 /mnt/mon_second_disque
```

Listing détaillé des fichiers dans le répertoire /mnt/mon_second_disque

```
root # ls -lh /mnt/mon_second_disque
```

```
total 88M
-rw-r-r-. 1 root root 88M 30 janv. 13:57 filesystem-defragmentation-...
```

Remplacement du disque 2 du *FS* par la partition /mnt/mon_second_disque

```
root # btrfs replace start -f 2 /dev/sdc /mnt/mon_second_disque/
```

Réparer un disque défectueux

Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs**
- Les sauvegardes

Conclusion

Affichage de l'état actuel des systèmes de fichier *btrfs*

```
root # btrfs filesystem show
```

```
Label: none uuid: b8e2b172-0e91-4854-a611-f1c98ad99638
```

```
Total devices 2 FS bytes used 76.09MiB
```

```
devid      1 size 149.00MiB used 148.00MiB path /dev/sdb2
```

```
devid      2 size 150.00MiB used 104.00MiB path /dev/sdc
```

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Exercice 4.4

Les backups

Différents types de pertes de données

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Différents types de pertes de données

Suppressions ou altérations (in)volontaires de fichiers

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Différents types de pertes de données

Suppressions ou altérations (in)volontaires de fichiers

- Conserver l'état des fichiers dans différentes versions.
- La sauvegarde doit être faite sur un système inaccessible à l'utilisateur.

Différents types de pertes de données

Suppressions ou altérations (in)volontaires de fichiers

- Conserver l'état des fichiers dans différentes versions.
- La sauvegarde doit être faite sur un système inaccessible à l'utilisateur.

Un bug logiciel au niveau du *FS* ou de l'*OS*

Différents types de pertes de données

Suppressions ou altérations (in)volontaires de fichiers

- Conserver l'état des fichiers dans différentes versions.
- La sauvegarde doit être faite sur un système inaccessible à l'utilisateur.

Un bug logiciel au niveau du *FS* ou de l'*OS*

- Avoir des sauvegardes indépendantes technologiquement avec le système à protéger.
- Exemple : ne pas se contenter de snapshots *Btrfs*, même s'ils sont copiés sur un autre disque.
- Avoir une sauvegarde la plus « basique » possible. Elle ne doit pas être réalisée dans un format supporté que par un petit nombre d'outils de restauration.

Différents types de pertes de données

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

Un accident physique

- Il faut le plus d'espace possible entre le système à sauvegarder et la copie. Idéalement, au moins 100 km.
- La sauvegarde ne doit pas être sur un support équivalent.
- Exemple : utiliser des marques de disque dur différentes de façon à ce qu'une faiblesse présente sur l'un d'eux n'ait que peu de chances de survenir sur l'autre au même moment.

Sauvegardes totales et incrémentales

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

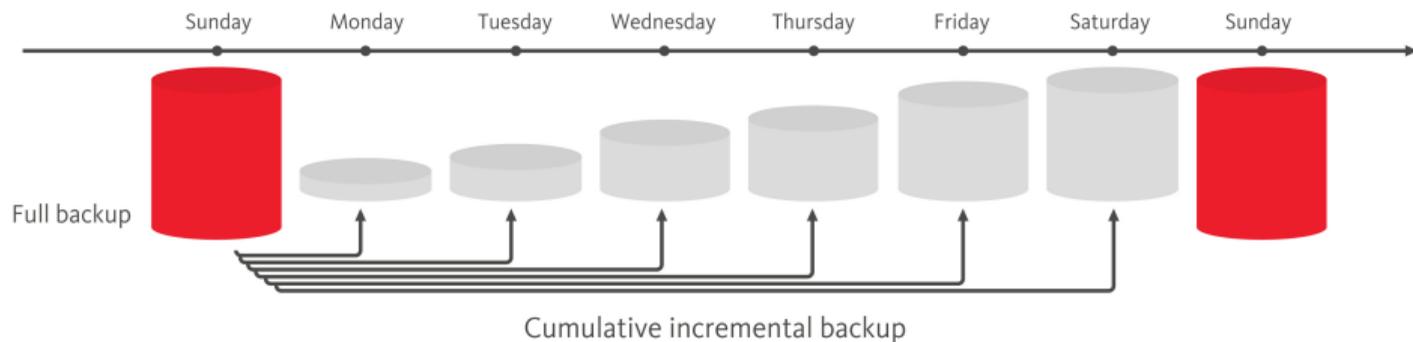
Le stockage

Btrfs

Les sauvegardes

Conclusion

Sauvegardes totales et incrémentales



Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

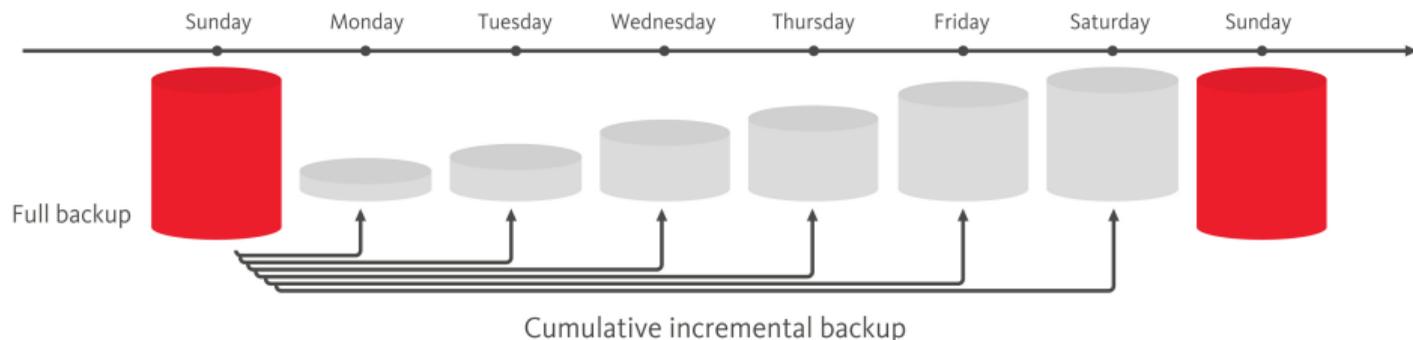
- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs
- Les sauvegardes**

Conclusion

Sauvegardes totales et incrémentales



- Le backup total copie à chaque fois l'intégralité des données.
- Le backup incrémental se contente des différences entre un état précédent et l'état actuel.
- En général, les solutions de backup utilisent les deux technologies ensemble, comme dans l'exemple ci-dessus.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion

- Il existe une multitude de supports de stockage différents : des disques durs, des disques réseau, de la mémoire en *RAM*, etc.
- Il existe aussi de nombreux systèmes de fichiers permettant l'utilisation de ces supports de diverses façons.
- La représentation physique réelle d'un fichier peut donc être très fragmentée et difficilement récupérable en cas d'incident. Mieux vaut miser sur les backups.
- Des pannes entraînant la perte de données peuvent survenir, il faut toujours être prêt à cette éventualité.

Introduction

Contexte

Le noyau

UNIX

Linux

Les distributions

Conclusion

Installation

Xubuntu

Les mises à jour

Gestion des
logiciels

Les bases

GNU

Le shell

Les utilisateurs

Les fichiers

Les processus

Les scripts

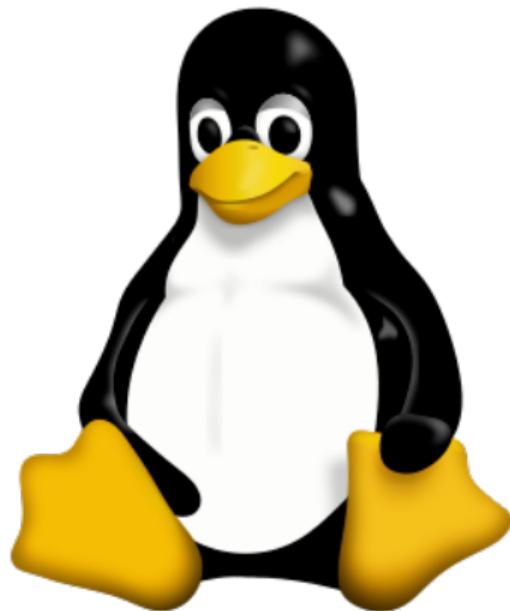
Les données

Le stockage

Btrfs

Les sauvegardes

Conclusion



Slides dispo à l'adresse :

https://share.spyzone.fr/cesi/infal51_presentation_tiu.pdf

Introduction

Contexte

- Le noyau
- UNIX
- Linux
- Les distributions
- Conclusion

Installation

- Xubuntu
- Les mises à jour
- Gestion des logiciels

Les bases

- GNU
- Le shell
- Les utilisateurs
- Les fichiers
- Les processus
- Les scripts

Les données

- Le stockage
- Btrfs
- Les sauvegardes

Conclusion



- Appel à mousser du *Linux User Group* de Strasbourg.
- Jeudi 9 avril, à partir de 19 h 00.
- La Taverne Des Serruriers, 25 Rue des Serruriers, Strasbourg.