

Systèmes UNIX. Introduction

Sergey Kirgizov

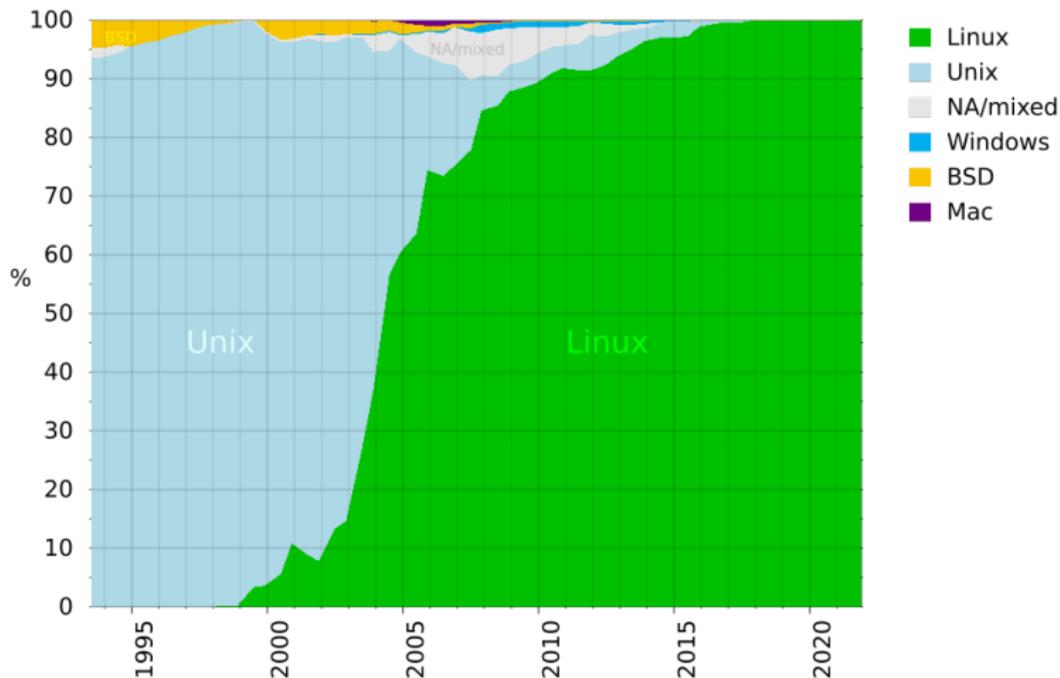
UNIX est partout !

par exemple chez vous...

- Installer un émulateur du terminal sur votre téléphone. Termux (Android) ou LibTerm (iOS).
- Execturer `uname -a` dans ce logiciel.
- `uname` c'est diminutif pour `unix name`.

Systèmes de type UNIX sont partout! sur vos portables, mais aussi sur les superordinateurs les plus puissants au monde

Les 500 superordinateurs les plus puissants au monde utilisent Linux.



Vous apprendrez :

- comment fonctionnent les systèmes d'exploitation UNIX
- comment sont-ils devenus si populaires et efficaces
- à utiliser l'interface homme-machine d'UNIX
- langages : Bash, expression régulières
- quelques notions d'administration UNIX

Vous allez créer vos propres logicielles d'UNIX :

- plusieurs scripts de traitement de données en TD
- un projet à faire en TP

Ce cours est fait en partie à partir du cours de Benoît Darties.
<https://benoit.darties.fr/>

Plan de cours

1. Histoire d'UNIX, son architecture
2. Fichiers, arbres, inodes, droits
3. Écosystème du shell, redirections, processus
4. Expressions régulières, grep et sed avancées
5. Shell comme langage de programmation
6. Réseau et UNIX

10 TD par groupe

4 TP par groupe — un projet à faire

Les notes : contrôle(s), projet à faire pendant les TD/TPs.

`https://kirgizov.link/teaching/esirem/unix`

Pré-requis

De la nécessité des systèmes d'exploitation

Caractéristiques des systèmes d'exploitation

Présentation d'UNIX

Pré-requis

Pré-requis

Pré-requis du cours

- Algèbre booléenne : et, ou , ou exclusif
- Convertir un nombre binaire en décimal
- Convertir un nombre décimal en binaire
- Puissances de 2 usuelles : 2, 4, 8, 16, 32, 64 , ...
- Dénombrement sur des mots binaires de taille n
- Bit, octet, notation hexadécimale d'un octet

Questions

- 1 Convertir 100100 en décimal
- 2 Convertir 47 en binaire
- 3 Combien de possibilités pour un mot binaire de 10 bits ?
- 4 Combien d'octet dans la notation hexadécimale AF EE 03 ?
- 5 Convertir 4A en binaire, puis en décimal.

Réponse :

Questions

- 1 Convertir 100100 en décimal
- 2 Convertir 47 en binaire
- 3 Combien de possibilités pour un mot binaire de 10 bits ?
- 4 Combien d'octet dans la notation hexadécimale AF EE 03 ?
- 5 Convertir 4A en binaire, puis en décimal.

Réponse :

Questions

- 1 Convertir 100100 en décimal
- 2 Convertir 47 en binaire
- 3 Combien de possibilités pour un mot binaire de 10 bits ?
- 4 Combien d'octet dans la notation hexadécimale AF EE 03 ?
- 5 Convertir 4A en binaire, puis en décimal.

Réponse :

- 1 $2^2 + 2^5 = 4 + 32 = 36$

Questions

- 1 Convertir 100100 en décimal
- 2 Convertir 47 en binaire
- 3 Combien de possibilités pour un mot binaire de 10 bits ?
- 4 Combien d'octet dans la notation hexadécimale AF EE 03 ?
- 5 Convertir 4A en binaire, puis en décimal.

Réponse :

- 1 $2^2 + 2^5 = 4 + 32 = 36$
- 2 $47/2 = 23[1]/2 = 11[1]/2 = 5[1]/2 = 2[1]/2 = 1[0]/2 = 0[1]$
Lecture à l'envers : 1 0 1 1 1 1

Questions

- 1 Convertir 100100 en décimal
- 2 Convertir 47 en binaire
- 3 Combien de possibilités pour un mot binaire de 10 bits ?
- 4 Combien d'octet dans la notation hexadécimale AF EE 03 ?
- 5 Convertir 4A en binaire, puis en décimal.

Réponse :

- 1 $2^2 + 2^5 = 4 + 32 = 36$
- 2 $47/2 = 23[1]/2 = 11[1]/2 = 5[1]/2 = 2[1]/2 = 1[0]/2 = 0[1]$
Lecture à l'envers : 1 0 1 1 1 1
- 3 $2^{10} = 1024$

Questions

- 1 Convertir 100100 en décimal
- 2 Convertir 47 en binaire
- 3 Combien de possibilités pour un mot binaire de 10 bits ?
- 4 Combien d'octet dans la notation hexadécimale AF EE 03 ?
- 5 Convertir 4A en binaire, puis en décimal.

Réponse :

- 1 $2^2 + 2^5 = 4 + 32 = 36$
- 2 $47/2 = 23[1]/2 = 11[1]/2 = 5[1]/2 = 2[1]/2 = 1[0]/2 = 0[1]$
Lecture à l'envers : 1 0 1 1 1 1
- 3 $2^{10} = 1024$
- 4 3 (2 caractères = $16 * 16$ possibilités = $256 = 2^8 = 1$ octet)
- 5 Binaire : 0100 1010. Décimal : $4 * 16 + 10 = 74$

De la nécessité des systèmes d'exploitation

Notion de Programme et Processus

Un programme

- Suite d'instructions interprétables par un ordinateur.
- Décrit dans un langage binaire et stocké dans un fichier.

Un processus

- Un processus est une instance d'un programme
- chargé dans la mémoire d'un ordinateur
- décrit l'état d'avancement d'un programme exécuté par l'ordinateur.
- Une fois les instructions entièrement exécutées, il disparaît.
- Pour 1 programme \Rightarrow 0, 1, ou n processus possibles.

Analogie : une recette de cuisine (programme), et l'action de cuisiner cette recette (processus)

Notion de Programme et Processus

Un programme

- Suite d'instructions interprétables par un ordinateur.
- Décrit dans un langage binaire et stocké dans un fichier.

Un processus

- Un processus est une instance d'un programme
- chargé dans la mémoire d'un ordinateur
- décrit l'état d'avancement d'un programme exécuté par l'ordinateur.
- Une fois les instructions entièrement exécutées, il disparaît.
- Pour 1 programme \Rightarrow 0, 1, ou n processus possibles.

Analogie : une recette de cuisine (programme), et l'action de cuisiner cette recette (processus)

Exécution d'un processus

Objectif d'un processus

- Exécuter les instructions du programme
- Manipuler le matériel (disques, imprimante, mémoire)
- Effectuer des calculs

Ressources requises

Un processus manipule des valeurs, stockées dans des variables et effectue des opérations sur ces dernières. Pour s'exécuter correctement, il a besoin de deux ressources primordiales :

- De la mémoire : stockage des données, des variables
- De la puissance de calcul : opérations sur les variables

Exécution d'un processus

Objectif d'un processus

- Exécuter les instructions du programme
- Manipuler le matériel (disques, imprimante, mémoire)
- Effectuer des calculs

Ressources requises

Un processus manipule des valeurs, stockées dans des variables et effectue des opérations sur ces dernières. Pour s'exécuter correctement, il a besoin de deux ressources primordiales :

- De la mémoire : stockage des données, des variables
- De la puissance de calcul : opérations sur les variables

Contrôle du matériel par un processus

Discussion processus - matériel

- Pour manipuler un périphérique, un processus pourrait directement "discuter" avec ce dernier, à condition qu'il sache comment dialoguer avec ce dernier
- Inconvénient : le processus doit savoir comment dialoguer avec le périphérique. Si le périphérique change, le processus doit alors apprendre à dialoguer avec le nouveau périphérique

Intérêt du système d'exploitation (SE) - *Operating System (OS)*

- l'OS sert d'interface entre le processus et le périphérique
- Le processus n'a plus besoin de savoir comment dialoguer avec le périphérique. S'il veut dialoguer avec un périphérique, il envoie son instruction au système d'exploitation, qui la communique au périphérique via son pilote (*driver*).

Contrôle du matériel par un processus

Discussion processus - matériel

- Pour manipuler un périphérique, un processus pourrait directement "discuter" avec ce dernier, à condition qu'il sache comment dialoguer avec ce dernier
- Inconvénient : le processus doit savoir comment dialoguer avec le périphérique. Si le périphérique change, le processus doit alors apprendre à dialoguer avec le nouveau périphérique

Intérêt du système d'exploitation (SE) - *Operating System (OS)*

- l'OS sert d'interface entre le processus et le périphérique
- Le processus n'a plus besoin de savoir comment dialoguer avec le périphérique. S'il veut dialoguer avec un périphérique, il envoie son instruction au système d'exploitation, qui la communique au périphérique via son pilote (*driver*).

Caractéristiques des systèmes d'exploitation

Rôle d'un système d'exploitation

Rôle principal

- Exécuter les processus : fournir les ressources nécessaires
- Assurer la liaison entre les ressources matérielles, les applications (processus), et l'utilisateur

Missions générales

- Gestion du processeur (ressource de calcul)
- Gestion de la mémoire
- Gestion des entrées-sorties (communication matériel)
- Exécution des applications
- Gestion des fichiers et droits d'accès
- Diagnostic de la machine

Question

- 1 Combien de systèmes d'exploitation connaissez vous ?
- 2 Combien existe-t'il de systèmes d'exploitation à votre avis ?

Les systèmes d'exploitation sont partout !

- Ordinateurs
- Serveurs, mainframes
- Périphériques réseau : commutateurs, routeurs, box internet
- PDA, Liseuses numériques
- Tablettes, smartphones
- Calculatrices
- Capteurs, TV connectées
- Même pour des lego (BrickOS)
- ...

Remarque

Des centaines de systèmes d'exploitation listés sur wikipedia (en) :

`https:`

`//en.wikipedia.org/wiki/List_of_operating_systems`

Caractéristiques d'un système d'exploitation

Gestion des utilisateurs

Un système d'exploitation peut être :

- Mono-utilisateur : un seul utilisateur du système (accès total)
- Multi-utilisateurs : plusieurs utilisateurs peuvent travailler
 - Notion de comptes utilisateurs
 - Besoin d'un espace personnel
 - Mécanismes de sécurité pour protéger la confidentialité

Caractéristiques d'un système d'exploitation

Gestion des processus

Un système d'exploitation peut être :

- Mono-tâche : un seul processus est exécuté à la fois
- Multi-tâches : plusieurs processus peuvent s'exécuter en simultané^a
 - Plus difficile à gérer
 - Besoin de mécanismes pour partager les ressources entre processus exécutés

a. en apparence, car en réalité : c'est plus compliqué

Questions

Pouvez-vous citer des exemples de systèmes

- 1 Mono-utilisateur et mono-tâche :
- 2 Mono-utilisateur et multi-tâches :
- 3 Multi-utilisateurs et multi-tâches :

Questions

Pouvez-vous citer des exemples de systèmes

- 1 Mono-utilisateur et mono-tâche : Calculatrice, MS-DOS
- 2 Mono-utilisateur et multi-tâches :
- 3 Multi-utilisateurs et multi-tâches :

Questions

Pouvez-vous citer des exemples de systèmes

- 1 Mono-utilisateur et mono-tâche : Calculatrice, MS-DOS
- 2 Mono-utilisateur et multi-tâches : iPads, Smartphones
- 3 Multi-utilisateurs et multi-tâches :

Questions

Pouvez-vous citer des exemples de systèmes

- 1 Mono-utilisateur et mono-tâche : Calculatrice, MS-DOS
- 2 Mono-utilisateur et multi-tâches : iPads, Smartphones
- 3 Multi-utilisateurs et multi-tâches : Windows 8, Mac OS X

Présentation d'UNIX

UNIX

Présentation d'UNIX

- Système d'exploitation multi-utilisateurs multi-tâches
- Créé en 1969 : plus de 50 ans d'histoire
- Référence dans le monde académique et l'industrie informatique
- Source d'inspiration pour nombreux OS : UNIX-like
- Forces :
 - possède un interpréteur de commande, le shell , qui permet une interaction poussée utilisateur - système
 - repose sur de nombreux petits utilitaires ayant chacun une action spécifique, et commutables entre eux
 - Forte interaction avec le langage C



Ken Thompson (gauche) - Dennis Ritchie (droite)

Multics machine



More info <https://www.multicians.org/>

La machine PDP-7.

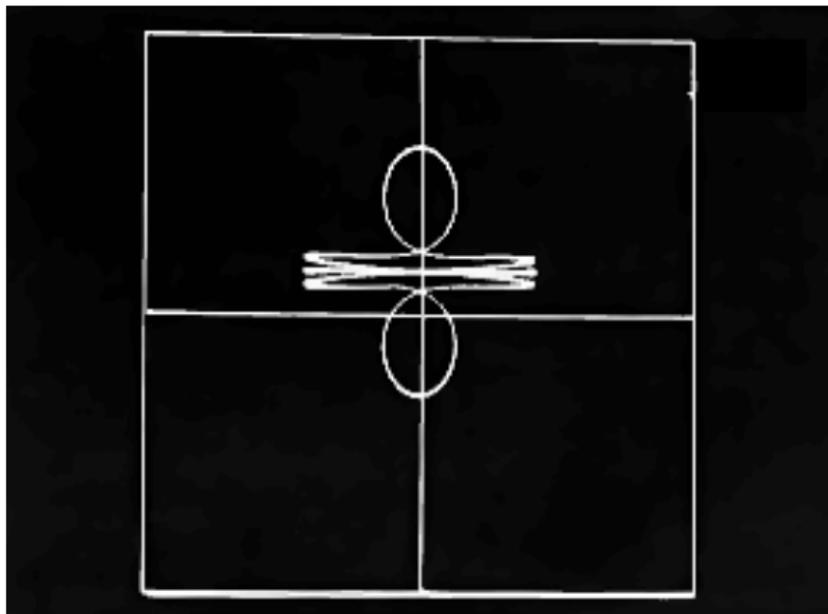


Les jeux video de l'époque

Spacewar!



Space Travel





UNIX : À l'origine du monde informatique #LPHDLT7



WeLoveDevs
4,01 k abonnés

S'abonner

👍 121



➦ Partager

✂️ Clip



<https://www.youtube.com/watch?v=Ab90tE1NbiQ>

Naissance d'UNIX

- 1964 Système Multics (MULTiplexed Information and Computing Service) par le MIT, Bell Labs et GE
- 1969 S'inspirant de Multics, Ken Thompson, ingénieur Bell Labs, crée un système d'exploitation mono-utilisateur en langage assembleur, qu'il baptise Unics
- 1970 Unics renommé en Unix, puis UNIX.
- 1971 Trop difficile à maintenir en langage assembleur. Ken Thompson définit le B, un nouveau langage de programmation inspiré du BCPL, dans l'optique de réécrire UNIX.
- 1972 Dennis Ritchie s'inspire du B pour définir un nouveau langage, le New B, renommé ensuite C
- 1972 UNIX entièrement réécrit en C

Naissance d'UNIX

- 1964 Système `Multics` (MULTiplexed Information and Computing Service) par le MIT, Bell Labs et GE
- 1969 S'inspirant de `Multics`, Ken Thompson, ingénieur Bell Labs, crée un système d'exploitation mono-utilisateur en langage assembleur, qu'il baptise `Unics`
- 1970 `Unics` renommé en `Unix`, puis UNIX.
- 1971 Trop difficile à maintenir en langage assembleur. Ken Thompson définit le B, un nouveau langage de programmation inspiré du BCPL, dans l'optique de réécrire UNIX.
- 1972 Dennis Ritchie s'inspire du B pour définir un nouveau langage, le New B, renommé ensuite C
- 1972 UNIX entièrement réécrit en C

Naissance d'UNIX

- 1964 Système `Multics` (MULTiplexed Information and Computing Service) par le MIT, Bell Labs et GE
- 1969 S'inspirant de `Multics`, Ken Thompson, ingénieur Bell Labs, crée un système d'exploitation mono-utilisateur en langage assembleur, qu'il baptise `Unics`
- 1970 `Unics` renommé en `Unix`, puis UNIX.
- 1971 Trop difficile à maintenir en langage assembleur. Ken Thompson définit le B, un nouveau langage de programmation inspiré du BCPL, dans l'optique de réécrire UNIX.
- 1972 Dennis Ritchie s'inspire du B pour définir un nouveau langage, le New B, renommé ensuite C
- 1972 UNIX entièrement réécrit en C

Naissance d'UNIX

1974 UNIX version 5. Depuis 1956, AT&T (dont dépend Bell Labs) ne peut commercialiser autre chose que des équipements téléphoniques. UNIX est alors donné avec code source aux universités (moyennant licence de faible coût)

1975 UNIX version 6. Première version à être commerciale autorisée.

1977 Branches de développement :

- La branche de recherche d'AT&T : UNIX versions 8 à 10
- La branche commerciale d'AT&T : System III, System V
- L'université de Californie à Berkeley : Berkeley Software Distribution jusqu'en 1994

1980's Apparition des UNIX-like (dont Linux), guerre des UNIX

Naissance d'UNIX

1974 UNIX version 5. Depuis 1956, AT&T (dont dépend Bell Labs) ne peut commercialiser autre chose que des équipements téléphoniques. UNIX est alors donné avec code source aux universités (moyennant licence de faible coût)

1975 UNIX version 6. Première version à être commerciale autorisée.

1977 Branches de développement :

- La branche de recherche d'AT&T : UNIX versions 8 à 10
- La branche commerciale d'AT&T : System III, System V
- L'université de Californie à Berkeley : Berkeley Software Distribution jusqu'en 1994

1980's Apparition des UNIX-like (dont Linux), guerre des UNIX

Naissance d'UNIX

1974 UNIX version 5. Depuis 1956, AT&T (dont dépend Bell Labs) ne peut commercialiser autre chose que des équipements téléphoniques. UNIX est alors donné avec code source aux universités (moyennant licence de faible coût)

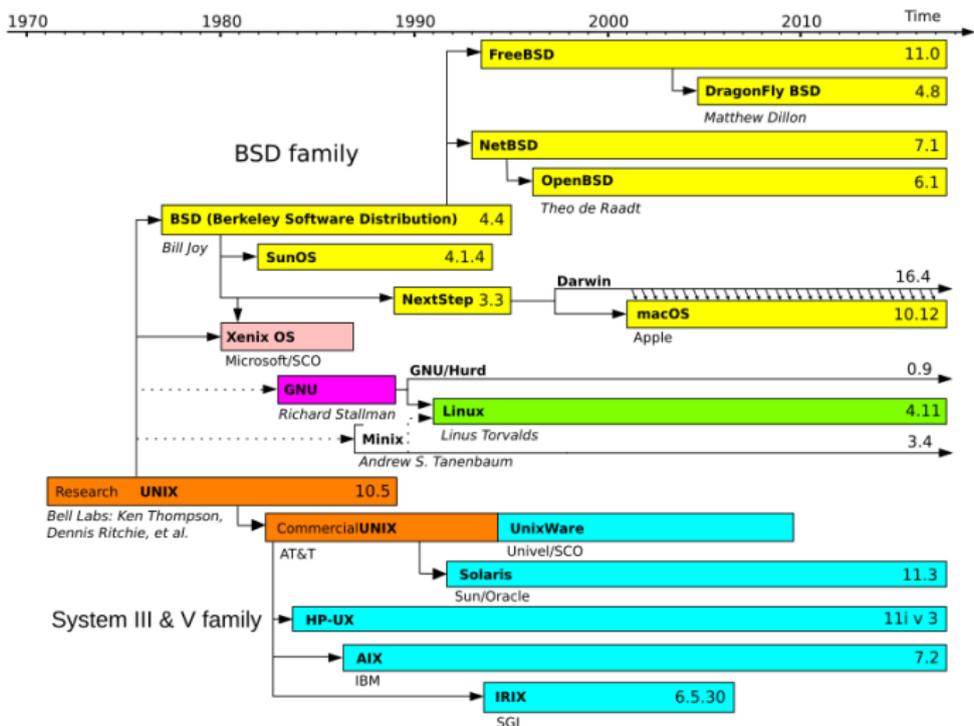
1975 UNIX version 6. Première version à être commerciale autorisée.

1977 Branches de développement :

- La branche de recherche d'AT&T : UNIX versions 8 à 10
- La branche commerciale d'AT&T : System III, System V
- L'université de Californie à Berkeley : Berkeley Software Distribution jusqu'en 1994

1980's Apparition des UNIX-like (dont Linux), guerre des UNIX

Unix Time Line



Questions ?