Systèmes d’exploitation

Fichiers et processus

Yves Stadler

Codasystem, UPV-M

13 mars 2013

Historique

Évolution d’UNIX

- 1969 Ken Thompson (Bell) version experimentale
- 1972 Dennis Ritchie créer le langage C
- 1973 réecriture du noyau et utilitaire en C
- 1974 distribution aux universités
- 1979 commercial

Historique

Évolution d’UNIX

- 1984 SystemV.2 devient un standard
- 1984 X/Open est chargé d’organiser la portabilité d’UNIX
- 1985 AT&T SVID (System V Interface Definition)
- 1986 SystemV.3 librairies partagées et Remote File System
- 1993 X/Open lance COSE (Common Open Software Environment) – Environement commun pour développement d’applications.

Historique

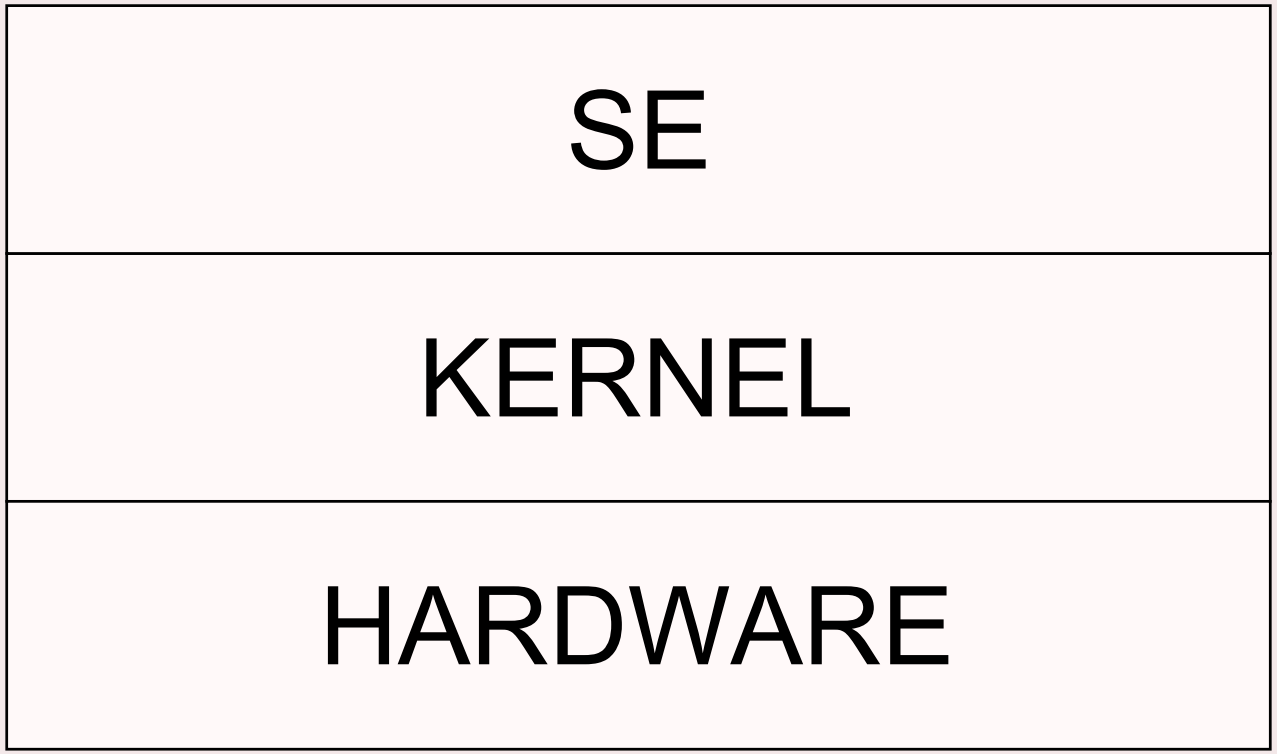
LINUX

- 1991 Linus Torvalds créer Linux a partir de MINIX
- 1994 Stable
- Aujourd’hui les versions 2 respecte les normes POSIX (Portable Operating System Interface X)

Organisation

Modèle en couche

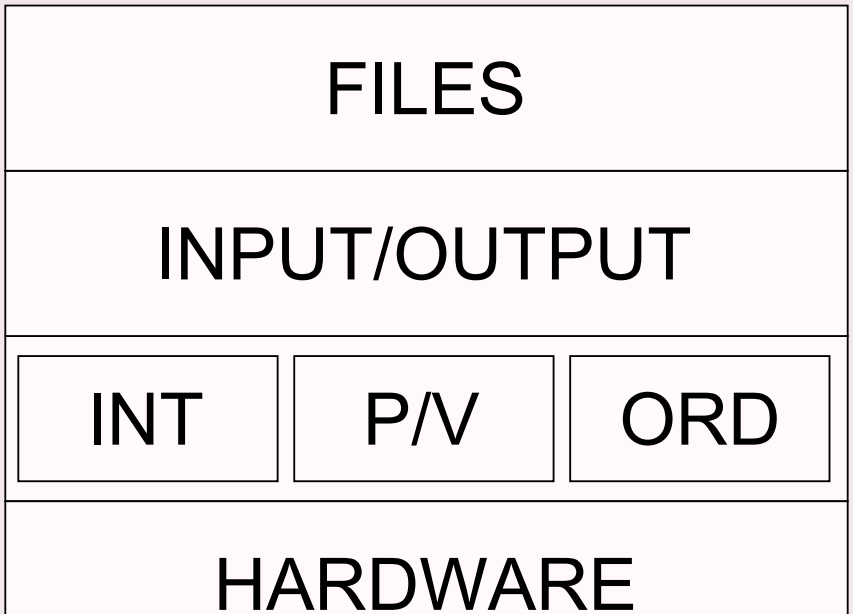
Le système peut se présenter comme un ensemble de couches.



Organisation

Modèle en couche

- Le noyau ou kernel constitue une interface entre le matériel est les programmes
- Il est lui même divisés en couches
- Il est irremplaçable par l'utilisateur
- Il gère les processus, les périphériques, la mémoire



Organisation

Descripteur de fichier

- Certaines caractéristiques (la date de création, de la dernière modification, le propriétaire, la taille) sont associés aux fichiers.
- Stockés dans un descripteur de fichier ou index-node ou i-node.
- Numéro

Gestion d'inodes

Le système garde à un ou des emplacements connus, une table de ses inodes.

Organisation

Les répertoires

Les répertoires sont une liste de (inode ; nom)

Les liens

Il existe deux types de liens :

- Hard link : c'est un pointeur sur un inode. ($\nleftarrow$ on ne peut pas changer de disque)
- Soft link : (lien symbolique) c'est un pointeur sur le nom d'un autre fichier.

La différence

- Les liens hards permettent d'utiliser le même contenu de fichier et d'en supprimer un sans détruire l'autre.
- Si le fichier source d'un lien symbolique est détruit, le lien deviens invalide.

Organisation

Gestion des droits

On utilise 12 bits pour enregistrer les droits d'accès :

2.1.0 others

5.4.3 group

8.7.6 user

9 Sticky : laisse le programme en mémoire pour un rappel plus rapide

10 GUID : donner les mêmes droits qu'au groupe

11 SUID : donner les mêmes droits qu'à l'utilisateur

Organisation

EUID et EGUID

Lorsque le bit SUID est activé le programme s'exécute avec l'UID du propriétaire du programme.

- Si l'UID du propriétaire est 0 : les programmes obtiens des droits root ! (ex : passwd)
- Si le programmeur n'est pas vigilant, cela peut-être une faille de sécurité (ex : si j'arrive à lancer un terminal avec un programme SUID root : j'aurai une console superutilisateur ! !)

Processus

Qu’est-ce qu’un processus

Activité associée a l’exécution d’un programme.

Ses caractéristiques

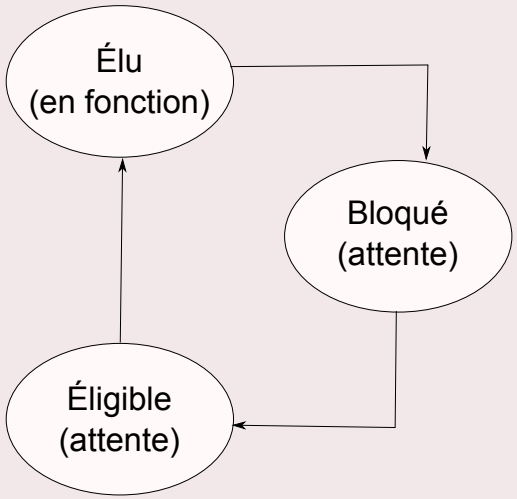
- Contexte (temps utilisé, zone u, piles, ...)
- État
- Espace de travail

Processus

Processus et processeur

Un seul processus peut avoir accès au processeur. Cela oblige certains à attendre le tour. Un processus peut-être :

- Élu (c'est lui qui utilise l'UC)
- Bloqué (il attent une entrée ou une sortie)
- Eligible (il n'est pas bloqué, mais n'as pas le processeur)

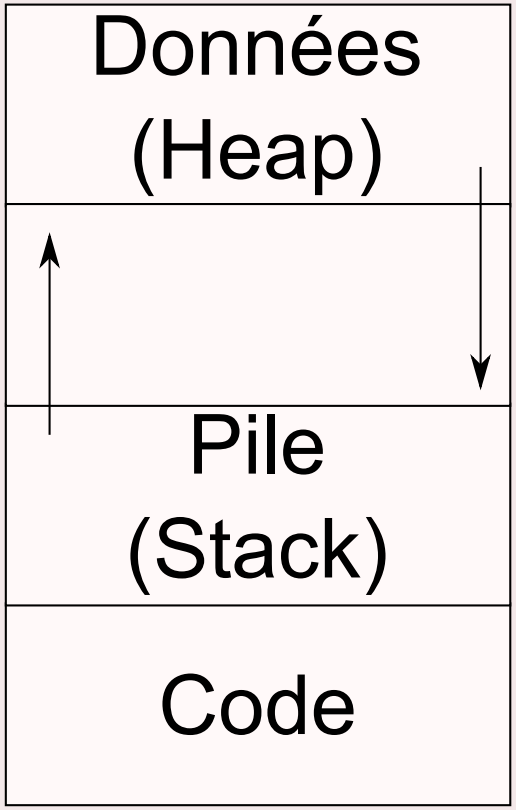


Processus

L’espace de travail

L’espace de travail est l’ensemble des données en mémoire nécessaires à l’exécution du processus.

- Le code : en langage ASM, liste des instruction pour le processeurs (Pointeur : Compteur Ordinal)
- Les données (le tas / heap) Ensemble des variables (globales/statiques/dynamiques) remplissage vers le haut
- La pile (stack) variables des appels de fonctions (inputs, valeur avant appel (co)...) remplissage vers le bas.



Processus

La zone u

Données privées du processus. Seule la zone u du processus en cours est manipulable. (struct user <sys/user.h>)
Son adresse se trouve dans le mot état.

- pointeur sur la structure de processus de la table des processus.
- uid réel et effectif
- Compteurs des temps (users et system) consommés
- Masque de signaux
- Terminal de contrôle du processus si celui-ci existe.
- Dernière erreur rencontrée pendant un appel système.
- Valeur de retour du dernier appel système.

Processus

La zone u

- E/S (les structures associées aux entrées-sorties)
- "." et "/" (le répertoire courant et la racine courante (c.f. chroot())
- La table des descripteurs
- imites de la taille des fichiers de la mémoire utilisable
- umask (masque de création de fichiers)

Processus

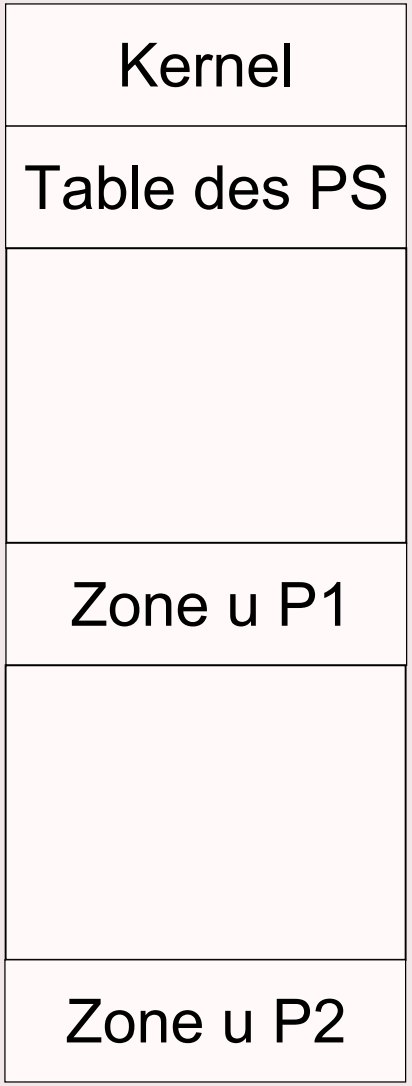
Contexte

- son état
- son mot d'état : en particulier (La valeur des registres actifs ; Le compteur ordinal)
- les valeurs des variables globales statiques ou dynamiques
- son entrée dans la table des processus
- sa zone u
- Les piles user et system
- les zones de code et de données.

Contexte

Lorsqu'un nouveau processus va être exécuté, il y a commutation du mot d'état et changement de contexte. Ces changements sont dictés par l'ordonnanceur.

Processus



Processus

Autres propriétés d’un processus

- PID
- Descripteurs de fichiers ouverts
- Priorité

Decripteurs de fichiers ouverts

- `0 <- /dev/term/c4` (stdin)
- `1 <- /dev/term/c4` (stdout)
- `2 <- /dev/term/c4` (stderr)
- `3 <- /tmp/toto`

Processus

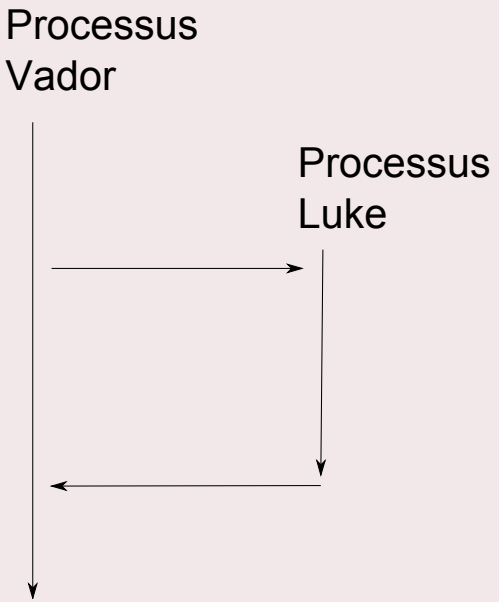
La différence entre programme et processus

Le processus est une activité d’un certain type qui possède un programme, des données en I/O et un état.

Processus

Hierarchie des processus

Tout processus qui en créer un autre devient un père. Le processus créer est son fils.
Le processus fils hérite de tout l'environement du père (variables, ...) sauf le PID et le PPID.



Processus

Hierarchie des processus

Les démons (ou daemon) sont des processus qui ne sont pas attachés à un terminal de contrôle. Ils sont souvent endormis. On peut voir tous les processus avec la commande : `ps -aux`

Hierarchie des processus

Les processus père et fils peuvent fonctionner en paralèlles, ou de façon asynchrone (le père attends que le fils se termine). En parallèle, il faut gérer les problèmes de synchronisation.