

Programmation Système

Introduction

Licence Sciences et Technologies Mention : Informatique

2ème Année

Références Bibliographiques (1)

Généralités sur les systèmes

- A. Tannenbaum, “Les Systèmes d’exploitation” (InterEditions)
- Silbetshatz, Galvin “Principes des systèmes d’exploitation”

Unix- mécanismes internes

- M.J. Bach, “Conception du système UNIX” (Masson/Prentice-Hall)
- Leffler & al, “Conception et implémentation du système 4.4BSD” (InterEditions)
- B. Goodheart & J. Cox, "The magic garden explained" (Prentice Hall)

Programmation système

- W. R. Stevens, “UNIX Network Programming” (Prentice-Hall)
- W.R. Stevens, “Advanced Programming in the UNIX Environment” (Addison Wesley)
- J.M. Rifflet, “La Programmation sous UNIX” (Mc Graw-Hill)
- J.M. Rifflet, “Les communications sous UNIX”

Références Bibliographiques (2)

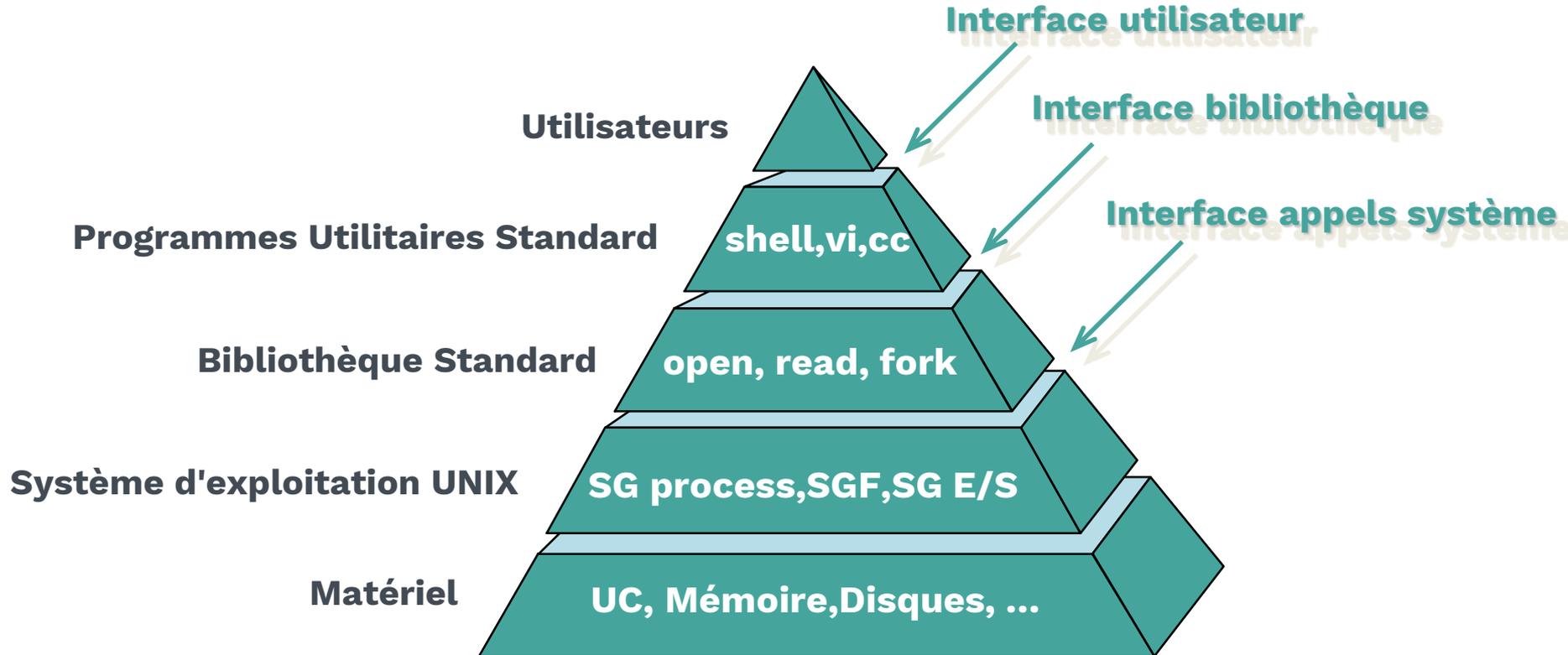
Administration

- E. Nemeth & al “UNIX System Administration Handbook”, (Prentice Hall)
- C. Hunt, "TCP/IP" (O'Reilly & Ass.)
- C. Bac, D. Bouillet, “Aministrer des systèmes en réseaux” (Dunod)

Références du langage C

- B. Kernighan, D. Ritchie “The C Programming Language” (Prentice Hall) -(Masson)
- S. Harbison, G. Steele “C : A Reference Manual”, (Prentice Hall) 3rd edition

U N I X : Un Système d'Exploitation ...



Les Objectifs d'UNIX

Simplicité et efficacité (par opposition aux gros systèmes MULTICS ...)

- Efficacité dans la gestion des ressources

Fournir des services d'exécution de programmes

- Charger, Exécuter, Gérer les erreurs, Terminaison
- Entrées / Sorties à partir de périphériques (Créer, Lire, Écrire, ...)
- Détecter les erreurs (CPU, mémoire, E/S, ...)

Fournir des services d'administration

- Allocation des ressources système
- Gestion des utilisateurs
- Comptabilité et statistiques
- Configuration
- Protection des ressources
- Ajout et retrait de gestionnaires de périphériques (drivers)

UNIX dans un Environnement Réparti

Services de gestion de données distantes

- Systèmes de fichiers répartis (RFS, NFS, ...)
- Transparence à la localisation, à la duplication
- Impression en réseau

Services de gestion d'utilisateurs distants

- Courrier Électronique, Transfert de fichiers
- Information sur les utilisateurs connectés

Services d'exécution distante

- Migration
- Gestion de la concurrence
- Parallélisation d'activités d'exécution
- Connexion à une machine distante

Les Points Forts d'Unix

Disponibilité du code source

- distribution du source et commercialisation par AT&T, nombreuses contributions, système ouvert (NFS, ...), ...

Simplicité

- small & simple, fichier = séquence d'octets, fichiers de configuration texte, ...

Interface uniforme des E/S

- tout est fichier, ensemble de commandes commun aux E/S, ...

Portabilité

- écrit en C, portage sur de nouvelles machines avec peu d'effort

Les Points Faibles d'UNIX

Interface utilisateur

- longtemps peu conviviale, orientée vers l'utilisation de terminaux alphanumériques

Multiplication des versions → situation chaotique

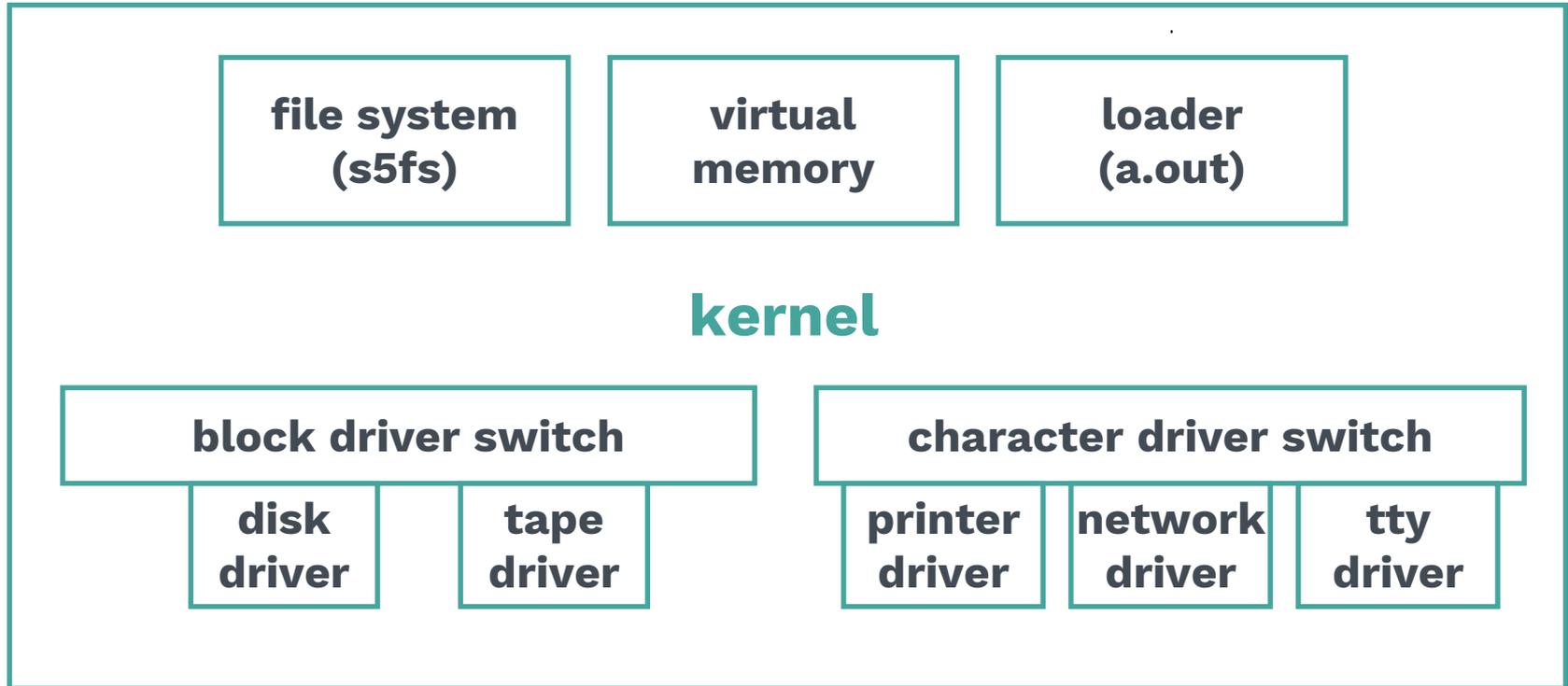
- prolifération incontrôlée, problème de compatibilité (vs AT&T et BSD), normalisation POSIX

Complexité de certaines versions → problèmes de robustesse

- ajout de code au noyau (important, non modulaire et complexe)

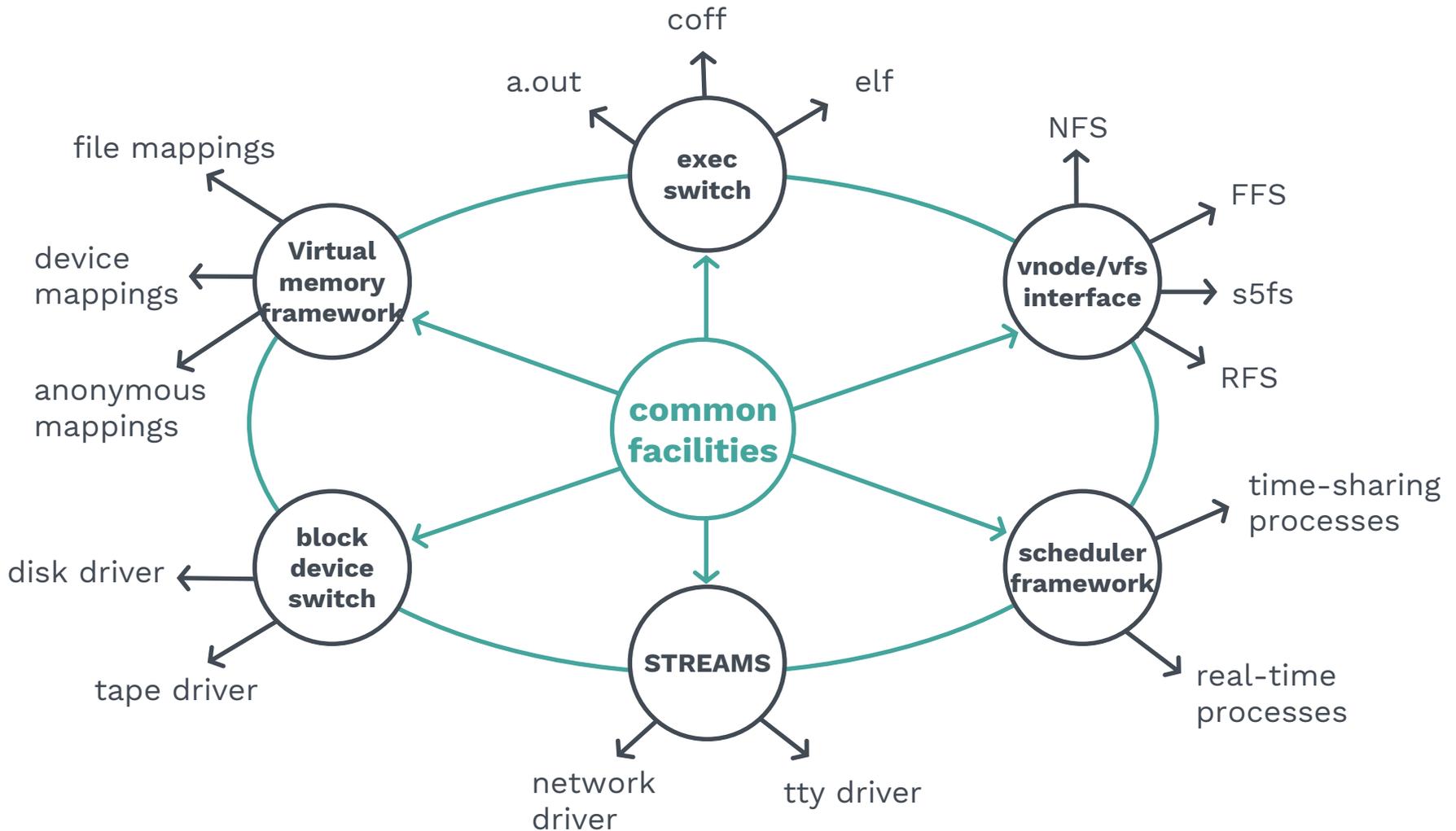
Approche monolithique → difficilement extensible

Noyau Unix Traditionnel



Uresh Vahalia, Unix Internals, Prentice Hall

Noyau Unix Moderne



Uresh Vahalia, Unix Internals, Prentice Hall

Caractéristiques d'UNIX

Système interactif en temps partagé, Multiutilisateurs et Multitâches

Principes

- Système de gestion de fichiers hiérarchisé
- Entrées/Sorties
- Création dynamique de processus (Père / Fils)
- Communication interprocessus (Pipes, Sockets, ...)

Langage de commande extensible (Shell)

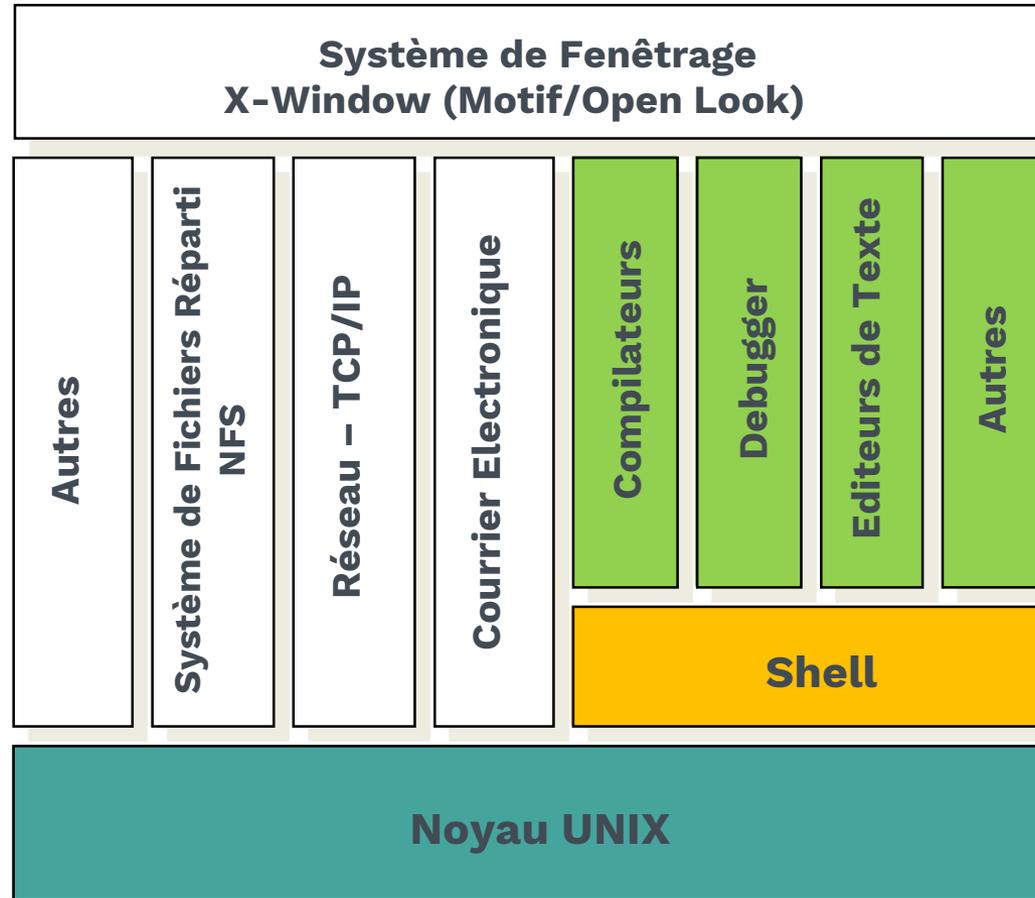
Noyau monolithique portable

- Le noyau est écrit en C à 95%.
- UNIX existe sur de nombreuses machines (PC, Stations RISC, CRAY-YMP, Hyper cubes, ...)

Nombreux outils dans la version de base du système

- Nombreux logiciels du domaine public

Architecture Générale Vue de l'Utilisateur



Programmes Utilitaires Standards

Editeurs

- ed, vi, emacs, sed, awk, ...

Compilateurs, Assembleurs, Éditeurs de liens

- cc, as, ld, F77, ...

Debuggers

- adb, sdb, dbx, gdb, ...

Gestionnaires de programmes et bibliothèques

- make, m4, ar, sccs, ...

Outils de communication

- mail, uucp, telnet, rlogin, talk, ftp, ...

Constructeurs de compilateurs

- lex, yacc, ...

Systeme de Fichiers

Organisation hiérarchisée arborescente

Accès uniforme aux données

- Fichiers de données
- Périphériques
- Mémoire

Protection

- un répertoire ne peut être modifié que par des primitives systèmes spécifiques

Droits associés à chaque fichier de l'arborescence

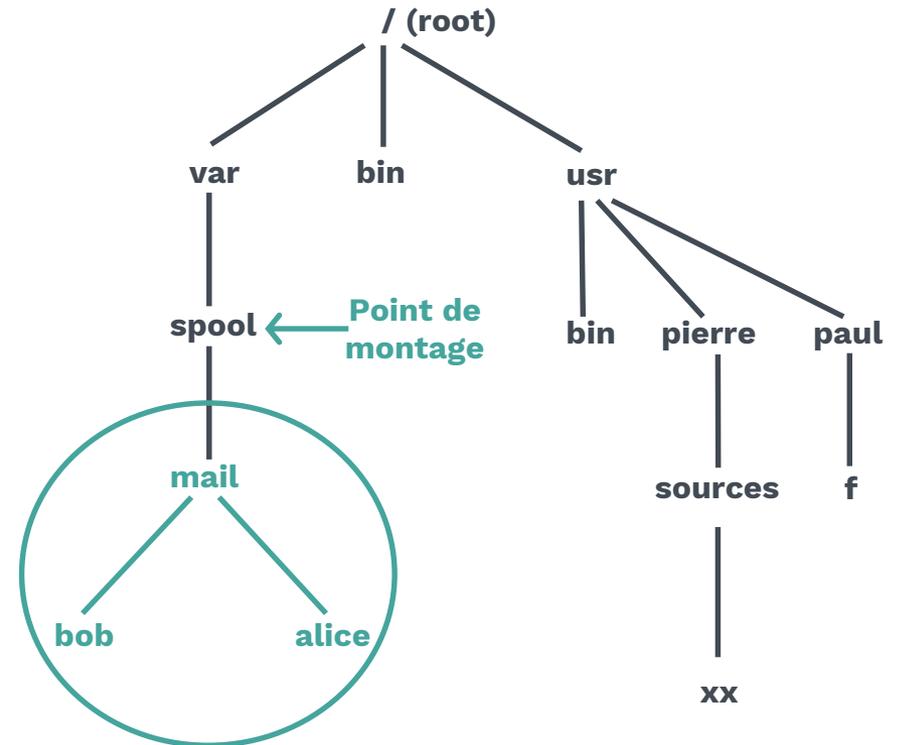
Systeme de fichiers virtuel dans les dernières versions d'UNIX (Virtual File System)

- Le système de gestion de fichiers est indépendant du type de machine, des types des disques, du cache disque, ...

Arborescence Standard

- `/` : le noyau UNIX
- `/bin` et `/usr/bin` : les commandes utilisateurs, quelques commandes d'administration du système
- `/tmp` : fichiers temporaires + swap.
- `/lib` et `/usr/lib` : les bibliothèques nécessaires à la programmation et à l'exécution des processus,
- `/etc` : les commandes d'administration du système + les fichiers de configuration.
- `/dev` : les fichiers spéciaux (device)

Possibilité de "montage"



Les Processus

Définition

- Entité dynamique représentant l'exécution d'un programme sur un processeur (programme en cours d'exécution)

Du point de vue du système d'exploitation

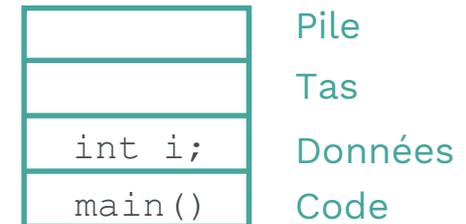
- Un espace mémoire confiné (code, piles, données)
- Code partagé
- état interne (compteur d'exécution, fichiers ouverts, ...)
- protection accès aux autres espace (autre tâche, noyau)

Programme et processus

Programme (code + données)

```
int i;  
int main() {  
    printf("Hello World");  
}
```

Processus



2 modes d'exécutions

- Utilisateur
- Système (appels système, interruptions)

Les Communications

Communications locales

- Tube, Fichier
- Mémoire partagée

Communications distantes

- Socket (TCP/UDP)
- STREAM (générique)
- Appel de procédure à distance (RPC) → plus de transparence

Unix n'est pas un système réparti !

Le Shell

Un processus interpréteur de commandes

- Intègre un langage (if, while, for, ...)
Possibilité de faire des programmes (scripts)
- Lancer/combiner des commandes
- Une commande = un processus

Algorithme du shell

- Lire une ligne
- Vérification syntaxique
- Si commande interne

Exécuter

Sinon

Création d'un processus exécutant le fichier associé
à la commande