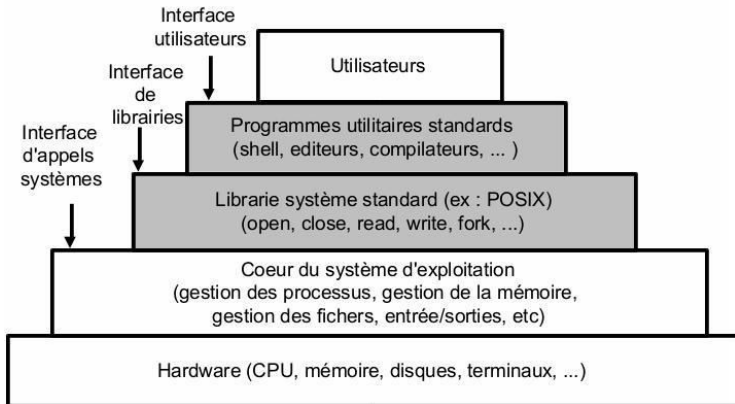


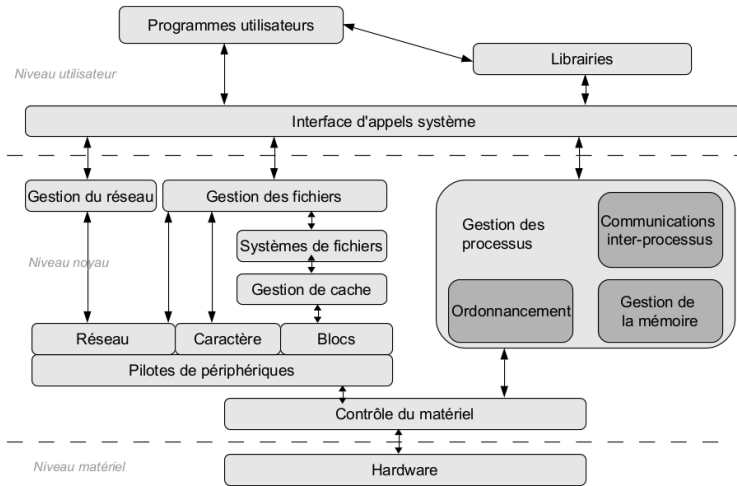
Les systèmes de la famille UNIX : Introduction

mickael.hoerd@hesge.ch

Structure et composants (1)



Structure et composants (2)



Fichiers (1)

Presque toutes les ressources système visibles sous la forme d'un fichier

- Un fichier n'est pas seulement un fichier stocké sur un disque dur ou autre.
- Un fichier est simplement un **flux d'octet nommé**, d'où qu'il vienne. (Souris, clavier, imprimante, caméra, disque dur, système de fichier).
- Le nom d'un fichier est unique à la machine et est déterminé dans un espace de nommage global hiérarchisé.

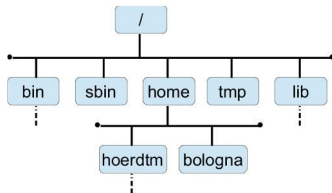
Fichiers(2)

Espace de nommage global hiérarchisé

Les données et les programmes sont organisés dans une **arborescence** de fichiers et de répertoires.

Notation

/	"racine"
/tmp	fichiers temporaires
/lib	librairies
/home	répertoires utilisateurs
/home/bologna	mon répertoire
...	



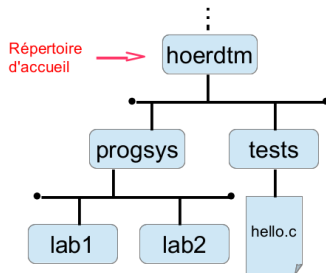
Fichiers(3)

Répertoire d'accueil

Les utilisateurs du système disposent d'un répertoire d'accueil appelé aussi *home directory*.

Environnement UNIX HEPIA

- Répertoire hébergé sur un serveur commun.
- Partagé entre toutes les machines en A406-07 et A433-32
- Protégé par des droits d'accès



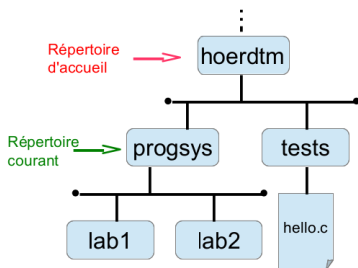
Fichiers(4)

Chemin d'accès

Un chemin d'accès ou *path* désigne la position d'un fichier (ou d'un répertoire) dans l'arborescence

Chemin d'accès relatif

- Un chemin d'accès **relatif** dépend du **répertoire de travail** courant (*working path directory*).
- Chemins d'accès spéciaux
 - Répertoire parent : ..
 - Répertoire courant : .
- Exemple: ../tests/hello.c



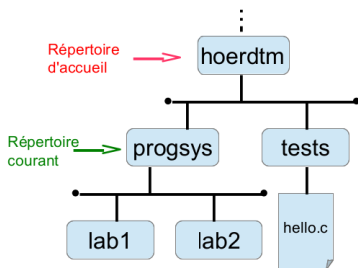
Fichiers(4)

Chemin d'accès

Un chemin d'accès ou *path* désigne la position d'un fichier (ou d'un répertoire) dans l'arborescence

Chemin d'accès relatif

- Un chemin d'accès **relatif** dépend du **répertoire de travail** courant (*working path directory*).
- Chemins d'accès spéciaux
 - Répertoire parent : ..
 - Répertoire courant : .
- Exemple: ../tests/hello.c



Fichiers(5)

Chemin d'accès absolu

Un chemin d'accès absolu est indépendant du répertoire de travail courant.

- Il se réfère à la racine de l'arborescence ("/")
- Exemples
 - /home/hoerdtm/tests/hello.c
 - /home/hoerdtm/progsys/lab1
 - ~/tests/hello.c
 - ~bologna/vision/vision.java
- ~ = Mon répertoire d'accueil ou celui d'un autre utilisateur.

Fichiers(6)

Attributs

Le SE stocke des méta-informations pour pouvoir organiser les ressources systèmes, on les appelle **attributs**.

Attributs de fichiers principaux

- Nom.
- Taille.
- Date de dernière modification
- Identifiant de l'utilisateur propriétaire.
- Identifiant de groupe propriétaire.
- Droits d'accès.

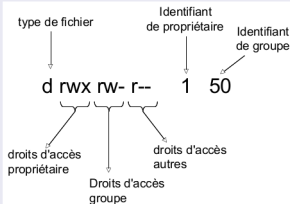
Fichiers(7)

Droits d'accès aux fichiers

UNIX gère les autorisations d'accès selon trois types d'utilisateurs, un super-utilisateur et trois type de permissions.

- Types de permissions : lecture, écriture et exécution.
- Types d'utilisateurs : Le propriétaire du fichier, les utilisateurs appartenant à un groupe bien défini, tous les autres.
- Le super utilisateur (*root*) : a tous les droits sur le système, indépendamment des permissions

Notation



Processus

Programme et processus

- Lorsqu'on démarre un programme, on crée un processus.
- Un programme est un fichier exécutable et un processus est une instance du programme en exécution.
- Plusieurs processus peuvent donc exécuter le même programme.
- Sous UNIX, plusieurs processus peuvent s'exécuter en parallèle.
- Les processus sont les entités principales actives dans un système UNIX.

Ne pas confondre un processus avec un programme!

Processus

Attributs de processus principaux

- Numéro unique : **PID** (Process IDentifier)
- Identifiant d'utilisateur réel (UID) : c'est l'identifiant de l'utilisateur ayant démarré le processus.
- Identifiant d'utilisateur effectif (EUID) : c'est l'identifiant d'utilisateur du processus pendant son exécution.
- Identifiant de groupe réel (GID) : c'est l'identifiant de groupe de l'utilisateur ayant démarré le processus.
- Identifiant de groupe effectif (EGID) : c'est l'identifiant de groupe du processus pendant son exécution.
- PID du processus parent
- Terminal d'attache pour les entrées/sorties/erreurs (*stdin, stdout, stderr*)

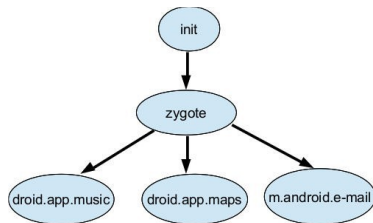
Processus

Arborescence de processus

Un processus est toujours créé par un autre processus appelé **processus parent**

- Tout processus a un processus parent sauf le tout premier.
- Le tout premier processus est appelé *init* et son PID est égal à 1

- Exemple d'arborescence de processus sous Android :



Processus

Processus et droits d'accès aux fichiers

- Les droits d'accès d'un processus aux fichiers s'appliquent comme suit :
 - Si l'EUID du processus est égal au propriétaire du fichier, c'est les droits d'accès utilisateurs qui s'appliquent.
 - Si l'EUID du processus n'est pas égal au propriétaire du fichier, mais que l'EGID est égal à l'identifiant de groupe du fichier, c'est les droits d'accès de groupe du fichier qui s'appliquent.
 - Sinon c'est les droits d'accès des autres qui s'appliquent.

Processus

Processus et droits d'accès aux fichiers

- Exemple :
 - Supposons un fichier dont les droits d'accès sont les suivants : `rwx r-x r- 1 50`
 - Un processus d'EUID 1 pourra lire, écrire et exécuter le fichier.
 - Un processus d'EUID 10 mais de EGID 50 pourra lire et exécuter le fichier.
 - Un processus d'EUID 10 et de EGID 100 ne pourra que lire le fichier.

Par défaut les droits d'accès dépendent de l'utilisateur qui exécute le programme, PAS de qui est propriétaire du fichier exécutable.

Processus

Lorsqu'un processus nécessite d'autres droits que l'utilisateur

- Un droit d'accès spécial existe, appelés "set user ID" et "set group ID"
- Lorsqu'un fichier a le droit d'accès posé, l'EID du process devient celui indiqué par le fichier, PAS celui de l'utilisateur qui exécute le programme.
- Exemple : la commande `/usr/bin/passwd`
- Notation, exécution = `s : rws rws —`
- Autre option : utilisation de la commande UNIX `su`.