

Ecole Nationale des Sciences Appliquées
Khouribga

Systeme d'exploitation Structure d'Unix

Med AMNAI
2018-2019

Introduction

1. Historique et généralités

- Composants d'un système informatique
- Historique des systèmes informatiques

2. Concepts de base d'Unix

- Le concept d'utilisateur
- Le concept de processus
- Le concept de fichier
- Structure générale d'Unix

3. Connexion et interfaces

- Le point de vue de l'utilisateur
- Le processus de connexion
- L'environnement graphique

4. Outils de base de Unix

- Xterm
- Emacs

Composants d'un système informatique

- Un **système informatique** comprend trois composants :
 - **l'ordinateur** proprement dit
 - les moyens de **communication**
 - le **système d'exploitation**

L'ordinateur (von Neumann)

- ❑ Ensemble physique comprenant :
 - **Processeur**
 - **Mémoire**
 - Des **organes** de communication (entrée et sortie)

Le système d'exploitation

- ❑ **Partie logicielle** du système informatique :
 - les programmes système:
 - les utilitaires (compilateurs, éditeurs, interpréteurs de commandes);
 - **le système d'exploitation** : ensemble de **programmes utilitaires**
 - les programmes d'application.
- ❑ **Gère** et **contrôle** les composants de l'ordinateur
- ❑ **Fournit une base** (machine virtuelle) sur laquelle seront construits les programmes d'application et les utilitaires:
services = {appels système}
- ❑ **Intermédiaire** entre utilisateur et matériel. Il gère l'utilisation de la totalité des **ressources** : temps, mémoire, fichiers, communications, etc;
- ❑ **But** :
Développer des applications sans se soucier des détails de fonctionnement et de gestion du matériel, ou des interactions entre les applications.

Fonctions principales d'un système d'exploitation

- ❑ **Gestion de périphériques**
- ❑ **Gestion de la mémoire**
- ❑ **Gestion de processeurs**
- ❑ **Gestion de processus, fils (threads) ou tâches**
- ❑ **Gestion de fichiers**
- ❑ **Protection et détection d'erreurs**

Concepts de base

- ❑ **Processus** : un programme en cours d'exécution, composé de: code + données + piles d'exécution, et de différents registres (e.g. compteur ordinal) caractérisant son état.
- ❑ **Fichiers** : ensemble de blocs de données stockés sur le disque
- ❑ **Mémoires virtuelles** : espaces d'adressage virtuels des processus (créés par les compilateurs).
- ❑ **Dispositifs d'entrée-sortie** : toute interaction avec le monde extérieur (clavier, souris, réseau).

Interface avec le matériel

- ❑ Chaque composant (processeurs, mémoires et périphériques) de l'ordinateur a son propre code (pilote ou driver) qui assure son fonctionnement et les interactions avec les autres.
- ❑ Le système d'exploitation gère et coordonne l'ensemble de ces composants notamment au moyen de lectures et d'écritures sur les bus, et d'interruptions.
- ❑ Les interruptions permettent au système d'exploitation de reprendre le contrôle:
 - Interruptions matérielles:
 - Horloges (pour gérer l'allocation des processeurs)
 - Périphériques (pour signaler la fin d'E/S)
 - Interruptions logicielles:
 - Erreurs arithmétiques
 - Données non disponibles en mémoire
 - Appels système (invocation du système d'exploitation)

Interactions utilisateur/système

- ❑ Les appels système peuvent être invoqués via un **interpréteur de commandes (shell)**, une **interface** graphique ou des **utilitaires**.
- ❑ L'interpréteur de commandes (Interface utilisateur/système) :
 - lancé dès la connexion au système ;
 - invite l'utilisateur à introduire une commande ;
 - récupère puis exécute la commande par combinaison d'appels système et d'outils (compilateurs, éditeurs de lien,...).
 - affiche les résultats ou les erreurs puis se met en attente de la commande suivante.

Sys. d'exploitation les plus représentés

- ▶ **Windows**, dominant sur le marché de l'ordinateur personnel
- ▶ **Mac-OS** d'Apple, système inséparable de son ordinateur, suivi de Mac-OS X (basé sur Unix (BSD))
- ▶ **UNIX**, seul système non lié à un constructeur ou un fabricant de logiciel, seul fonctionnant sur tout ordinateur, sous diverses formes
- ▶ **GNU/Linux**, s'appuie sur le noyau Unix (différentes distributions disponibles : **Debian**, **Ubuntu**, **CentOS**, **Red Hat**, **Fedora**, **Mandriva**, **SuSE** ...)

2. Concepts de base de Unix

- Le concept d'utilisateur
- Le concept de processus
- Le concept de fichier
- Structure générale de Unix

Le concept d'utilisateur

- ▶ sur un **ordinateur personnel** :
l'utilisateur ne s'identifie pas si l'ordinateur est en libre service, l'utilisateur doit transporter ses propres données. Les fichiers présents sur l'ordinateur sont à tout le monde et n'appartient à personne.
- ▶ avec **Unix** :
chaque utilisateur doit s'identifier, ses fichiers lui appartiennent et il peut en autoriser ou interdire l'accès par les autres, ses données sont contenues dans son **répertoire personnel**.

La procédure de connexion

- ▶ L'utilisateur est désigné par son **nom d'utilisateur** et un **mot de passe**;
- ▶ La procédure de **connexion vérifie ces deux informations**, et n'accepte qu'un utilisateur dument enregistré;
- ▶ Quand l'utilisateur se **déconnecte**, il libère l'ordinateur;
- ▶ L'ordinateur **fonctionne en permanence**, on ne l'arrête pas;
- ▶ Le mécanisme de **partage des fichiers** *permet à l'utilisateur* d'atteindre ses données depuis n'importe quel ordinateur relié au **serveur de fichiers**.

Attributs de l'utilisateur

- ▶ Nom d'utilisateur;
- ▶ Mot de passe : est conservé sous forme cryptée ; c'est l'élément fondamental de la sécurité;
- ▶ Répertoire personnel : sa place dans la hiérarchie des fichiers est déterminée par l'administrateur ; l'utilisateur ne peut normalement **placer ses fichiers** qu'ici;
- ▶ Programme de démarrage : programme avec lequel l'utilisateur dialogue au démarrage de la connexion ; la fin de ce programme termine la connexion;
- ▶ Groupe(s) d'appartenance.

À propos du mot de passe

Vous devez respecter quelques règles :

- ▶ ne l'écrivez nulle part;
- ▶ choisissez une chaîne facile à retenir, mais ne figurant dans aucun dictionnaire;
- ▶ incluez dans cette chaîne au moins un signe de ponctuation;
- ▶ incluez dans cette chaîne au moins un chiffre mélangez majuscules et minuscules;
- ▶ n'utilisez jamais de caractères accentués;
- ▶ apprenez à taper votre mot de passe rapidement.

Le super-utilisateur

L'administrateur a les privilèges du **super-utilisateur**; il peut :

- ▶ Lire et modifier tout les fichiers sur le système ;
- ▶ Enregistrer les nouveaux utilisateurs et initialiser leur environnement de travail ;
- ▶ Supprimer un utilisateur ;
- ▶ Installer ou mettre à jour des logiciels ;
- ▶ Surveiller le bon fonctionnement du système et corriger les défauts ;
- ▶ Effectuer les sauvegardes périodiques des programmes et données.

Le concept de processus

- ▶ Un **processus** est un programme en cours d'exécution;
- ▶ La plupart des commandes **exécutent un programme** et donc lancent un processus;
- ▶ Le programme de démarrage correspond à un **processus présent pendant toute la session**;
- ▶ Des dizaines ou centaines de processus sont en fonctionnement à tout moment.

Le concept de processus

Caractéristiques d'un processus

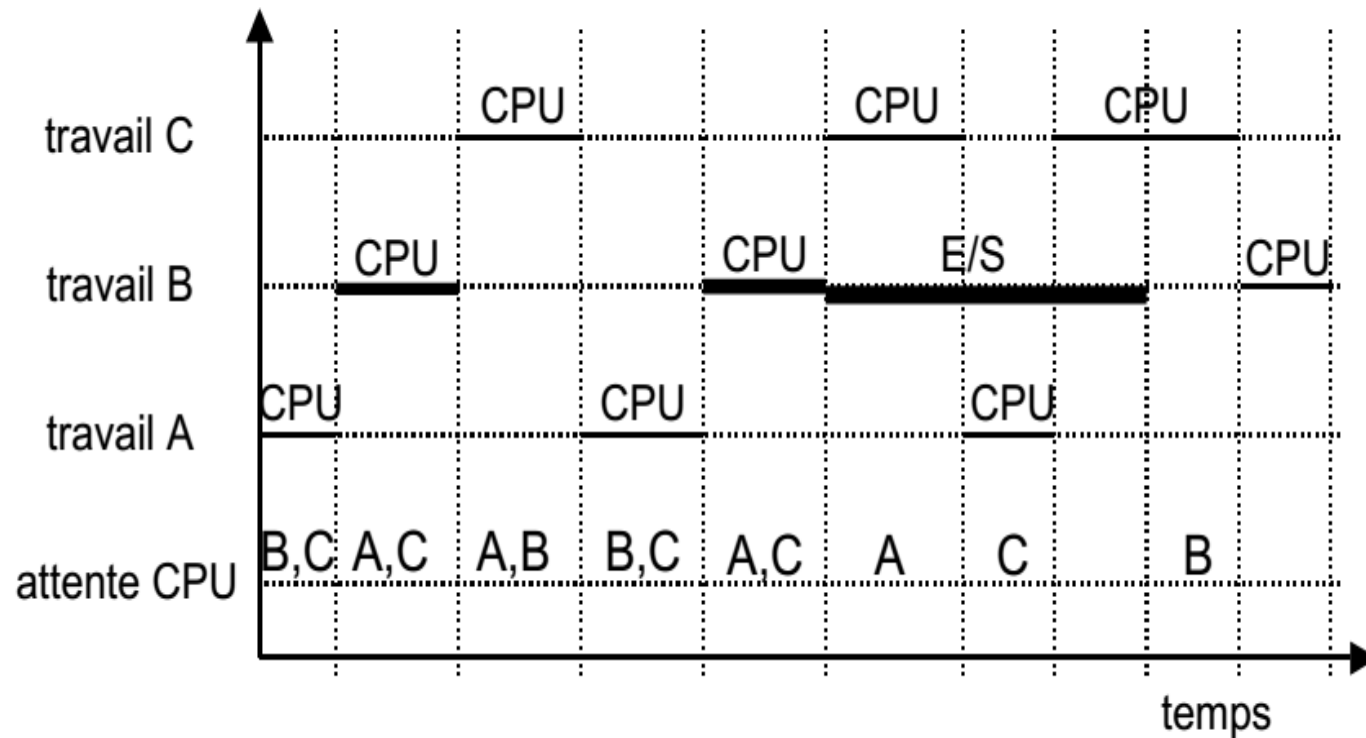
- ▶ **identifié** par un numéro entier ;
- ▶ **code** du programme en cours d'exécution ;
- ▶ **données** traitées par ce code ;
- ▶ identification des **fichiers en cours** de traitement et leur état ;
- ▶ **répertoire courant** ;
- ▶ identité du **propriétaire** du processus ;
- ▶ **terminal** associé.
- ▶ etc.

Le concept de processus : États

- ▶ Les processus sont dans différents états :
 - En attente d'un **événement extérieur** (action de l'utilisateur) ;
 - En attente **d'exécution** (tranche de temps) ;
 - En attente de l'arrivée d'une partie de **la mémoire virtuelle** ;
 - En **exécution** (un seul à la fois).
- ▶ Tout processus est lancé par un processus père :
 - **Arbre généalogique** des processus ;
 - **Propriétaire réel** (utilisateur qui l'a lancé) ;
 - **Propriétaire effectif** (utilisateur donnant les droits du processus).

Le concept de processus : États

► Simulation (CPU, trois tâches ABC)



Le concept de fichier

- ▶ Toutes les informations extérieures au processus sont des fichiers ;
- ▶ Un fichier peut être associé au clavier, à l'écran, à l'imprimante, etc.
- ▶ Quatre catégories de fichiers :
 - **Ordinaires**
 - **Répertoires**
 - **Spéciaux**
 - **Liens symboliques**

Le concept de fichier

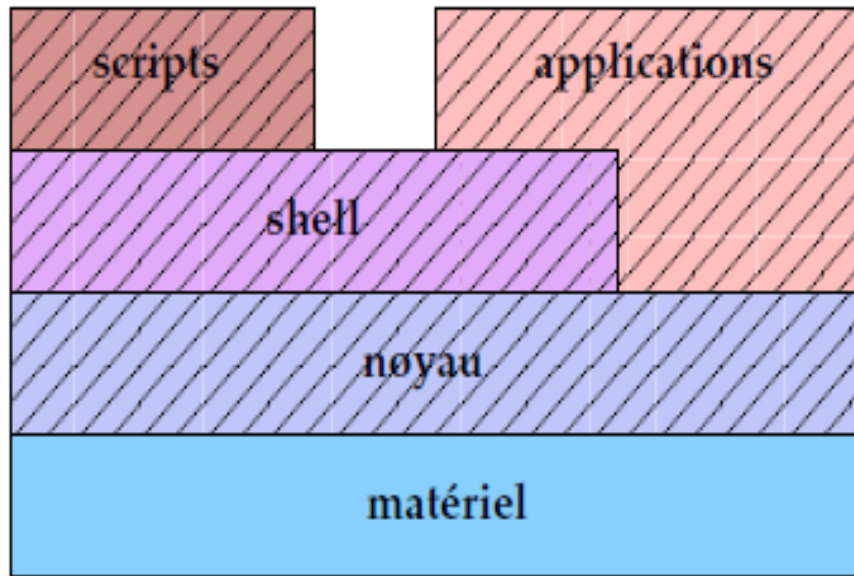
- ▶ **Fichier ordinaire (.txt, .exe,...):**
 - Suite d'octets sans structure particulière;
 - Contient des données ou des programmes;
 - Fichiers de texte structurés en lignes par une marque de fin.
- ▶ **Répertoire :**
 - Noeud de la hiérarchie des fichiers;
 - Fichier de références à d'autres fichiers.

Le concept de fichier

- ▶ **Fichier spécial** (périphériques, ..) :
 - Fichier **virtuel**, représentation d'un **organe périphérique** ;
 - accès par **un programme pilote**, spécifique du périphérique.
- ▶ **Lien symbolique** :
 - Fichier contenant **la chaîne de caractères** qui représente le **nom d'un autre fichier** ;
 - Moyen de **référence indirecte** ;
 - Moyen de construire un **graphe quelconque** et plus seulement une arborescence.
- ▶ **Cheminement** dans la hiérarchie :
 - le passage d'un répertoire à un autre se note /
 - le répertoire **racine** s'appelle seulement /

Structure générale d'Unix

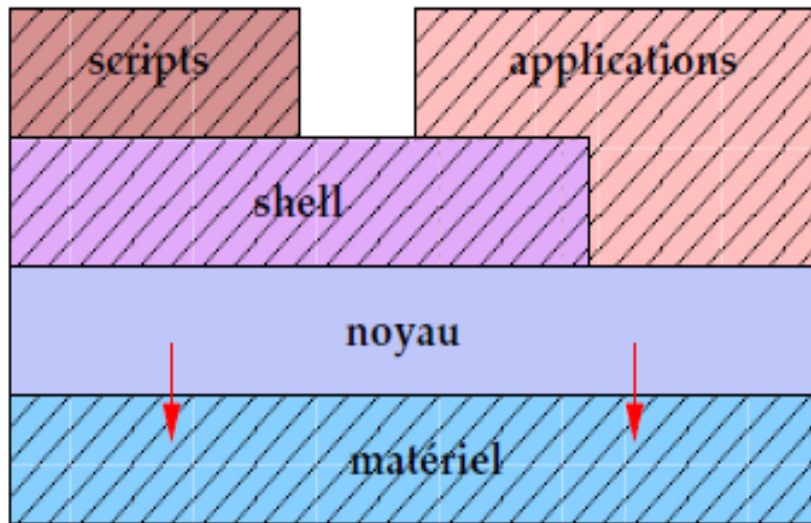
Meilleure représentation



- **Le matériel** est au niveau le plus bas :
le processeur son langage propre (langage machine)

Structure générale d'Unix

Le noyau (kernel)



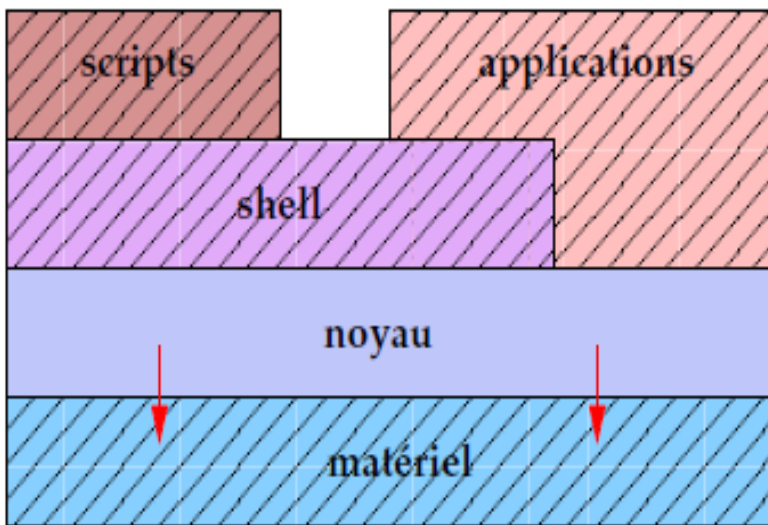
- ▶ le **noyau** d'Unix masque le matériel on n'accède aux ressources du matériel que par les **opérations primitives** (open, close, read, write, fork,...).

cela comprend :

- gestion du **système de fichiers**
- partage du **temps** du processeur ;
- partage de la **mémoire** ;
- accès aux **périphériques** grâce aux **pilotes**.

Structure générale d'Unix

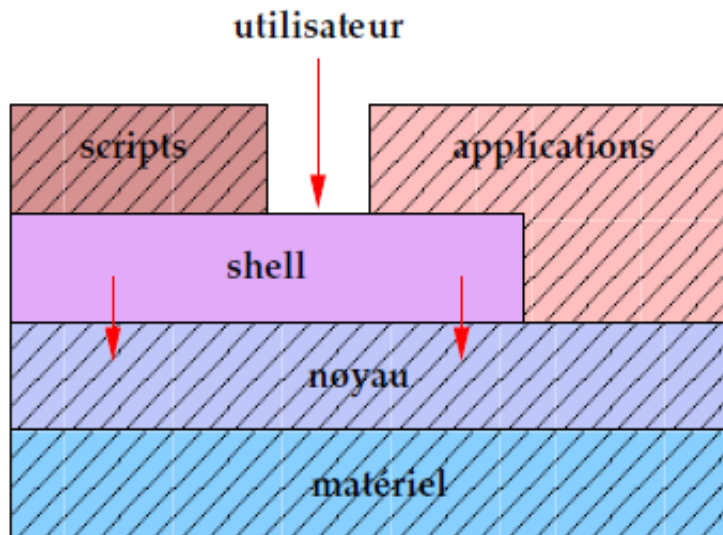
Le noyau (kernel)



- ▶ Accès aux primitives par instructions d'appel au système ;
- ▶ Les primitives s'exécutent en **mode privilégié** ;
- ▶ Le reste des programmes est en **mode utilisateur** ;
- ▶ Les primitives permettent de :
 - lancer des **processus** ;
 - lire ou écrire sur **des fichiers** ;
 - obtenir de la place **en mémoire** ;
 - etc.

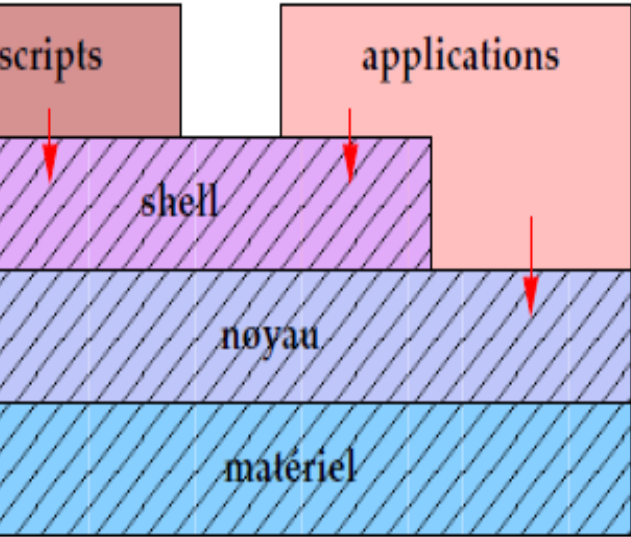
Structure générale d'Unix

Le shell

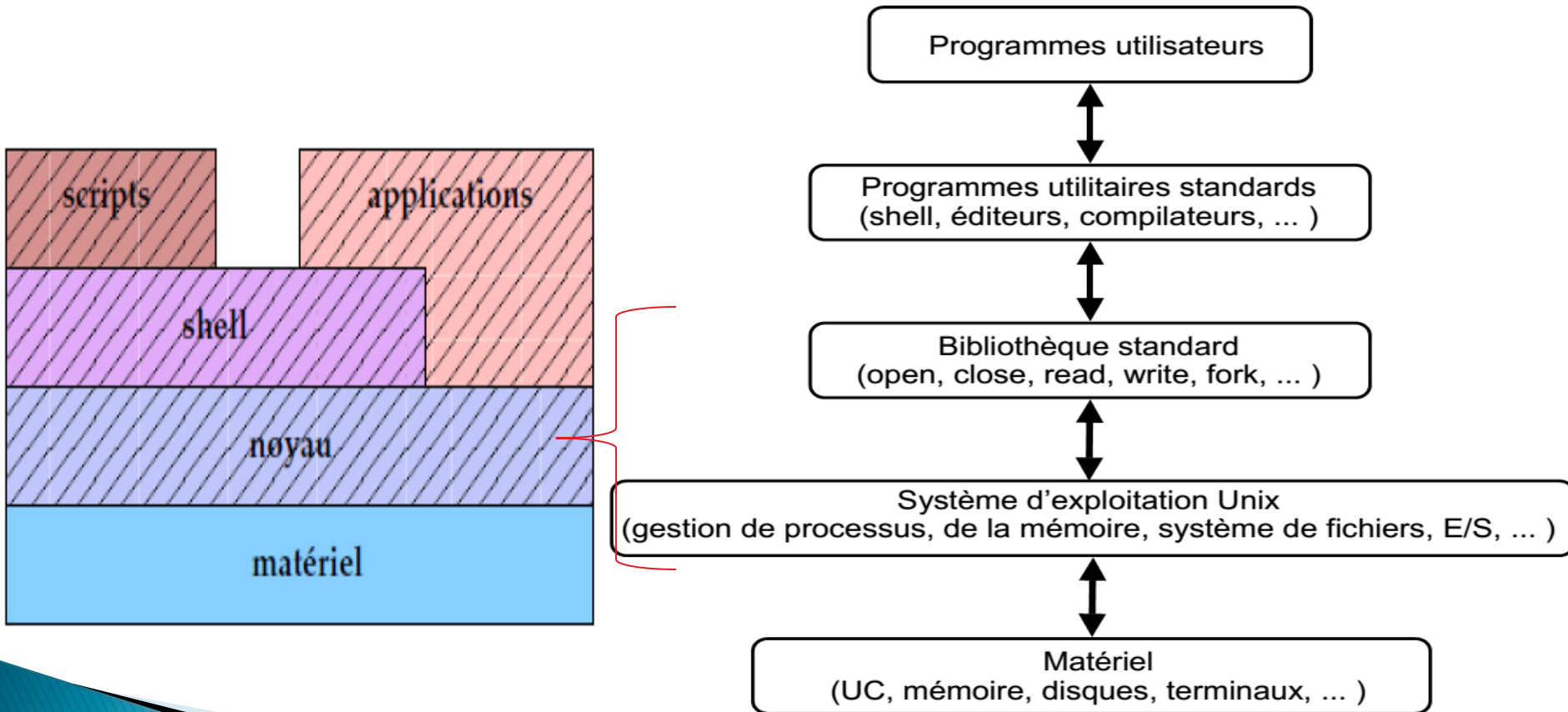


- ▶ le programme de démarrage qui interprète les commandes est le **shell** (coquille de l'amande en anglais);
- ▶ le shell est « à l'écoute » de l'utilisateur ;
- ▶ il **interprète** et **exécute** les commandes tapées ;
- ▶ quand le processus appelé par la commande se termine, le processus du **shell redevient actif**.

Structure générale d'Unix

- Scripts et applications**
- 
- The diagram illustrates the layers of the Unix system architecture. It consists of three stacked horizontal layers, each with a different color and diagonal hatching pattern. The top layer is light blue with diagonal lines and contains two boxes: 'scripts' on the left and 'applications' on the right. The middle layer is light green with diagonal lines and contains the label 'shell'. The bottom layer is light red with diagonal lines and contains the label 'noyau'. The bottom-most layer is light yellow with diagonal lines and contains the label 'matériel'. Red arrows indicate the flow of communication: one arrow points from the 'scripts' box down to the 'shell' layer, another from the 'applications' box down to the 'shell' layer, and a third from the 'applications' box down to the 'noyau' layer.
- ▶ la plupart des programmes d'application communiquent avec le noyau sans passer par le shell ;
 - ▶ le shell est un langage directement interprétable (langage de script) ;
 - ▶ on peut utiliser ce langage pour construire des scripts.

Structure générale d'Unix



Structure générale d'Unix

Types de shells existants

- ▶ le shell est indépendant du noyau ;
- ▶ il existe plusieurs shells plus ou moins perfectionnés :
 - **sh**, shell de Steven Bourne, conçu au début d'Unix, beaucoup de scripts l'utilisent ;
 - **csh**, shell de la première version BSD, de syntaxe proche de celle de **C** ;
 - shells perfectionnés dérivés des précédents :
 - **ksh**, dérivé de **sh** ;
 - **tcsh**, dérivé de **csh** ;
 - **bash**, version améliorée de **sh** ;
 - **zsh**, qui englobe tous les autres.

3. Connexion et interfaces

- Le point de vue de l'utilisateur
- Le processus de connexion
- L'environnement graphique

Le point de vue d'utilisateur

- ▶ l'ordinateur lui-même est l'hôte
- ▶ les utilisateurs se connectent à un hôte donné à l'aide d'un terminal :
 - Terminal alphanumérique, aujourd'hui simulé par une fenêtre de l'outil XTERM (ou une de ses variantes);
 - Terminal graphique, également appelé terminal X;
 - Station de travail ou ordinateur personnel, où les composants du terminal graphique sont indissociables de l'ordinateur;

Le point de vue d'utilisateur

Interface alphanumérique

- ▶ l'interface alphanumérique est celle d'une fenêtre de Xterm ;
- ▶ **un seul processus** peut communiquer avec l'interface (clavier et affichage, pas de souris);
- ▶ le processus **attaché au terminal** est *interactif, à l'écoute* des commandes tapées par l'utilisateur;
- ▶ commandes sous forme de **suites de caractères** ;
- ▶ on peut lancer un processus **détaché du terminal**, qui passe *en arrière-plan*.

Le point de vue d'utilisateur

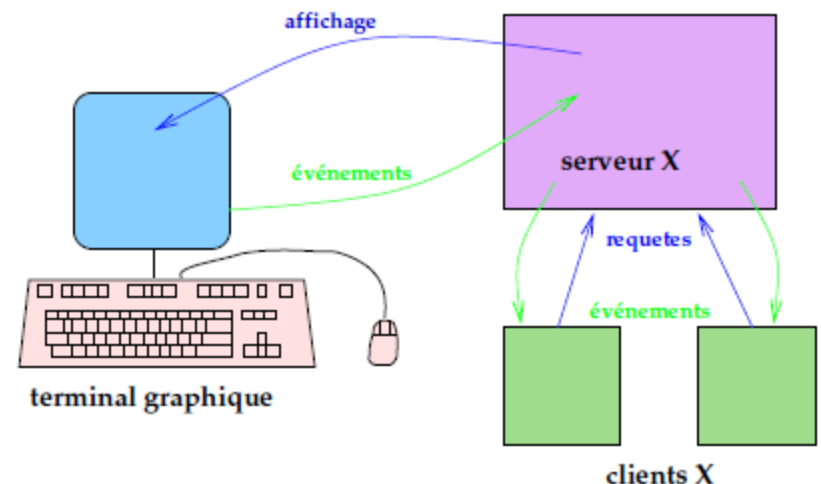
Interface graphique

- ▶ l'interface graphique nécessite un système de fenêtrage;
- ▶ celui qu'on utilise avec Unix s'appelle **X** ou **X11**;
- ▶ idée fondamentale s'appuyant sur la relation client-serveur :
- le **serveur X** gère le terminal graphique dans sa totalité :
 - Affichage sur l'écran graphique;
 - Reconnaissance des signaux (événements) envoyés par le clavier et la souris;
 - Reconnaissance d'événements graphiques (passage de la souris dans une fenêtre, recouvrement d'une fenêtre par une autre, etc.).
- Les clients sont des programmes qui envoient au **serveur X** des requêtes d'affichage et reçoivent la notification des événements qui les concernent.

Le point de vue d'utilisateur

Le serveur X

- ▶ le système de fenêtrage est **indépendant des machines**;
- ▶ le même serveur X peut satisfaire des requêtes provenant de **plusieurs machines** ;
- ▶ les clients ne savent pas comment fonctionne le serveur, et *vice-versa*.



Le point de vue d'utilisateur

Le gestionnaire de fenêtres

- ▶ le système X n'impose **aucun comportement particulier** aux clients ;
- ▶ **l'interface graphique** n'est pas imposée (contrairement à Windows ou Mac-OS) :
décor , présence de **menus** déroulants ou **apparence** de boutons ou **icônes** traitement possible des **fenêtres** etc.
- ▶ tout cela est réalisé par un client particulier, le **gestionnaire de fenêtres** ;
- ▶ plus récemment on a ajouté par au-dessus un **environnement de bureau**, qui codifie des comportements et des apparences.

Le processus de connexion

Connexion par interface graphique

- ▶ L'écran d'accueil sera celui de Gdm ;
- ▶ En milieu d'écran apparaît la fenêtre de dialogue ;
- ▶ Saisie du **nom d'utilisateur** et du **mot de passe** :
 - Le système vérifie **l'adéquation des deux informations** ;
 - Si elle est bonne, la **session commence** ;
 - S'elle ne l'est pas, le système **ne dit pas pourquoi** (par sécurité).

Le processus de connexion

Démarrage de la session

- ▶ Une fois l'identification faite, le système fait démarrer :
 - Le **serveur X**, (c'est-à-dire le **système de fenêtrage**)
 - Le **gestionnaire de fenêtres** (ici METACITY) ;
 - **L'environnement de bureau** (ici GNOME) ;
 - Un ou plusieurs **tableaux de bord**, qui rassemblent les moyens graphiques de communication ;
 - Un ou plusieurs **clients X**, c'est-à-dire des applications d'utilisation fréquente ;
- ▶ Tout ceci constitue **la configuration de la session** ;
- ▶ On peut la modifier pendant toute la session.

Le processus de connexion

Connexion par interface textuelle

- ▶ La connexion par interface textuelle sert dans de nombreuses circonstances :
 - changement d'identité sur la même machine;
 - connexion à une machine différente mais proche;
 - connexion à distance.
- ▶ On est dans une interface textuelle locale, typiquement une fenêtre **Xterm** ;
- ▶ Une commande permet de lancer la connexion, nous la verrons plus tard ;
- ▶ Le système demande successivement le nom d'utilisateur et le mot de passe (dialogue avec un shell).

L'environnement graphique

Le tableau de bord

on doit y trouver :

- ▶ Le changeur de bureau, qui permet de changer d'écran virtuel ;
- ▶ Quelques boutons de lancement des applications les plus fréquentes ;
- ▶ La liste des fenêtres du bureau visible ;
- ▶ Un phylactère explicatif (une infobulle) s'ouvre quand le pointeur passe dessus ;
- ▶ On le paramètre facilement par le menu accessible par le bouton droit de la souris.

4. Outils de bas de UNIX

- Xterm
- Emacs

Xterm

- ▶ **XTERM** est un **client graphique** simulant un terminal alphanumérique.
- ▶ Il existe beaucoup d'outils de même nature, exp : **Gnome** ;
- ▶ **GNOME-TERMINAL** est aussi complet, avec des perfectionnements de présentation ;
- ▶ Dans la fenêtre, un **shell** est **à l'écoute** :
 - **lit et interprète** les commandes saisies ;
 - **affiche les résultats**.
- ▶ **Élargir la fenêtre** est presque toujours inutile ;
- ▶ **L'allonger en hauteur** est souvent intéressant.

Xterm

Mode de fonctionnement de Xterm

- ▶ Le caractère saisi au clavier est envoyé au processus, qui envoie l'écho dans la fenêtre.
- ▶ La touche **Ctrl** retransmet **64** au code de la touche enfoncée en même temps :
 - **C-a** envoie le code **0** (zéro)
 - **C-g** correspond au signal auditif et annule en général ce qui est en cours
 - **C-j** est la fin de ligne
 - **C-m** est le retour, noté **RET**

Xterm

Saisie des commandes

- ▶ La ligne saisie n'est envoyée au shell qu'après appui sur la touche **RET** (touche **Entrée**)
- ▶ Cette touche peut être tapée **n'importe où** dans la ligne
- ▶ Tant qu'elle n'est pas tapée on **peut corriger** la ligne
 - **Déplacements** par les touches **←** et **→**
 - **C-a** amène en **début** de ligne, **C-e** en **fin** de ligne
 - **C-w** efface le **mot précédent**, **C-k** tout **ce qui suit le curseur**,
 - **C-u** toute la ligne
 - **C-c** **abandonne** la commande en cours de saisie

Xterm

Édition de texte

- ▶ Il existe des **éditeurs spécialisés**, intégrés dans une application;
- ▶ Il existe aussi **des éditeurs universels**, qui peuvent **travailler sur tout type de fichier**
- ▶ Les éditeurs les plus simples ne font que cela : **ED, EX, VI, VIM, XEDIT, NANO, etc.**
- ▶ **EMACS** est un véritable **éditeur universel** :
 - véritable **environnement** de **programmation** ;
 - fonctionne sous **tout système** ;
 - logiciel **libre** ;
 - facile à **étendre** et **adapter** ;
 - très riche.

Apprentissage de Unix

Difficultés d'apprentissage de Unix

- ▶ L'apprentissage de Unix est long et difficile :
 - Rechercher l'information;
 - Apprendre à se servir des outils de recherche;
 - Pas d'ordre logique d'apprentissage;
 - Nécessité d'être rapidement opérationnel;
 - Nécessité de revenir souvent sur la plupart des points.
- ▶ Évolution de Unix par accumulation, entraînant beaucoup de redondance.

Référence

Livre:
LINUX
Initiation et utilisation
2^e édition
Dunod