



Eléments d'Informatique

Cours 4 - Système d'Exploitation: fichiers, processus Compilation

Catherine Recanati

UNIVERSITÉ PARIS 13
NORD

Plan général

- Représentation des nombres. Notion de variable.
- Programme. Expressions.
- Architecture des ordinateurs: langage machine, langage assembleur, AMIL.
- **Systemes d'exploitation : fichiers, processus, compilation.**
- Instructions de contrôle: boucles et branchements.
- Programme, définition de fonction, appel fonctionnel.
- Tableaux de variables et fonctions d'arguments de type tableau.
- Sens d'un programme, pile d'exécution, compilation.
- Pointeurs et tableaux.
- Chaines de caractères, bibliothèque <string.h>.
- Allocation dynamique, liste chaînées.
- Révisions.



- Cours 4 -
*Systeme d'exploitation:
fichiers, processus
Compilation*

- Systeme d'exploitation
- Fichiers, processus
- Compilation

Plan

Systeme
d'exploitation

Fichiers

Processus

Compilation

Systeme d'exploitation

Un **systeme d'exploitation** (SE) est un ensemble de programmes permettant et facilitant l'exploitation d'un ordinateur.

Le SE (ou OS) permet aux applications de se partager les ressources de l'ordinateur comme le processeur (temps partagé), la memoire, les peripheriques (imprimante, clavier, ecran, souris, etc.). Les interactions avec le SE se feront par l'intermediaire de notions abstraites comme les **fichiers**, les **processus** ou celle d'**utilisateur**.

Plan

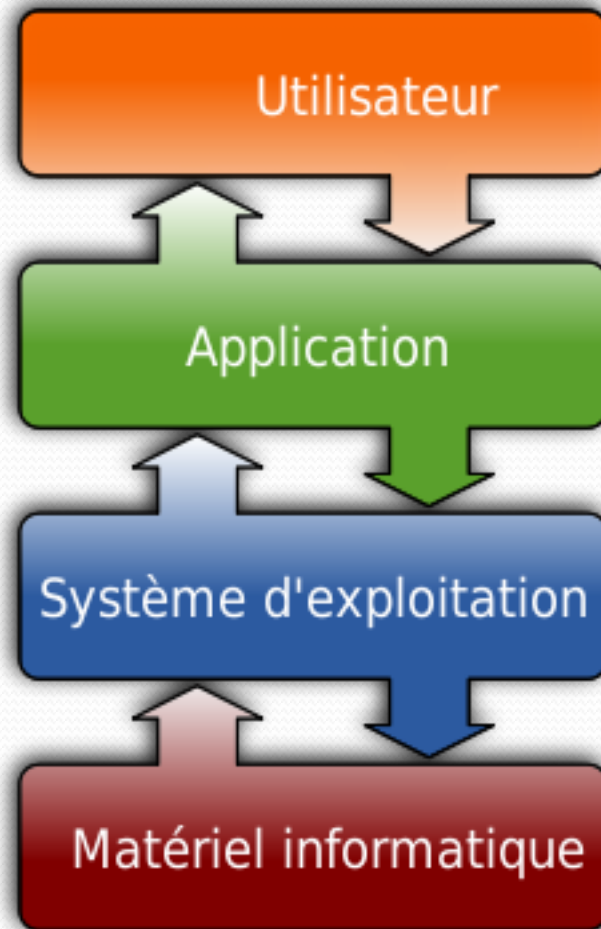
Système d'exploitation

Fichiers

Processus

Compilation

Systeme d'exploitation



Un SE permet à plusieurs utilisateurs ou processus (programmes) de se partager les ressources de l'ordinateur: exécuter des programmes, lire et écrire des informations, manipuler des fichiers, détecter des erreurs, mais aussi communiquer via le réseau entre ordinateurs.

Plan

Systeme
d'exploitation

Fichiers

Processus

Compilation

Systeme d'exploitation

Le systeme d'exploitation est lance lors de la mise en marche de l'ordinateur dans une phase dite d'amorçage (*bootstrap*).

On peut faire tourner differentes versions d'un SE sur un meme ordinateur, mais bien entendu, pas simultanement. La version utilisee determine les fonctionnalites accessibles aux applications et a l'utilisateur.

Plan

Systeme
d'exploitation

Fichiers

Processus

Compilation

Systeme d'exploitation

Il existe sur le marche des dizaines de systemes d'exploitation differents mais il n'y a en realite que deux grandes familles: **Unix** (dont Mac OS X, GNU/Linux, iOS et Android) et **Windows**.

Bien que moins bien conu qu'Unix, Windows detient le quasi-monopole sur les ordinateurs personnels (avec pres de 90% de part de marches depuis 15 ans).

UNIX est un systeme d'exploitation **multi-tache** et **multi-utilisateur** creé aux Laboratoires Bell en 1969 par Kenneth Thompson.

annee universitaire: 2016 -2017

Plan

Systeme
d'exploitation

Fichiers

Processus

Compilation

Systeme d'exploitation

A noter que **le langage C** a aussi été inventé dans les Laboratoires Bell (en 1972). Il a été développé en même temps qu'UNIX par Dennis Ritchie et Ken Thompson pour permettre l'écriture des programmes du système.



année universitaire: 2016 -2017

Plan

Système d'exploitation

Fichiers

Processus

Compilation

Unix

Unix repose sur **un interpréteur de commandes**, le **shell** - qui permet de lire et d'interpréter des commandes sur une ligne de commande - et sur de nombreux petits utilitaires, les commandes, réalisant des tâches spécifiques, et pouvant être appelées depuis la ligne de commande de l'interpréteur.

Ainsi, dans une fenêtre « console » ou « terminal » de votre environnement de travail, tourne un shell interpréteur de commandes Unix. Parmi ces commandes figure la commande `ls`, qui permet de lister les fichiers, et la commande `rm` (*remove*), qui permet de les supprimer.

année universitaire: 2016 -2017

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Fichiers

Les **fichiers** sont des zones mémoires servant à stocker des textes, des codes, des images ou encore des dossiers contenant des noms de fichiers (répertoires).

En fait, la notion de fichiers est plus large, et le SE permet de manipuler les entrées du clavier, les sorties sur la console ou sur une imprimante comme des fichiers.

Les fichiers sont organisés en arbres et un gestionnaire de fichiers permet de visualiser leur arborescence.

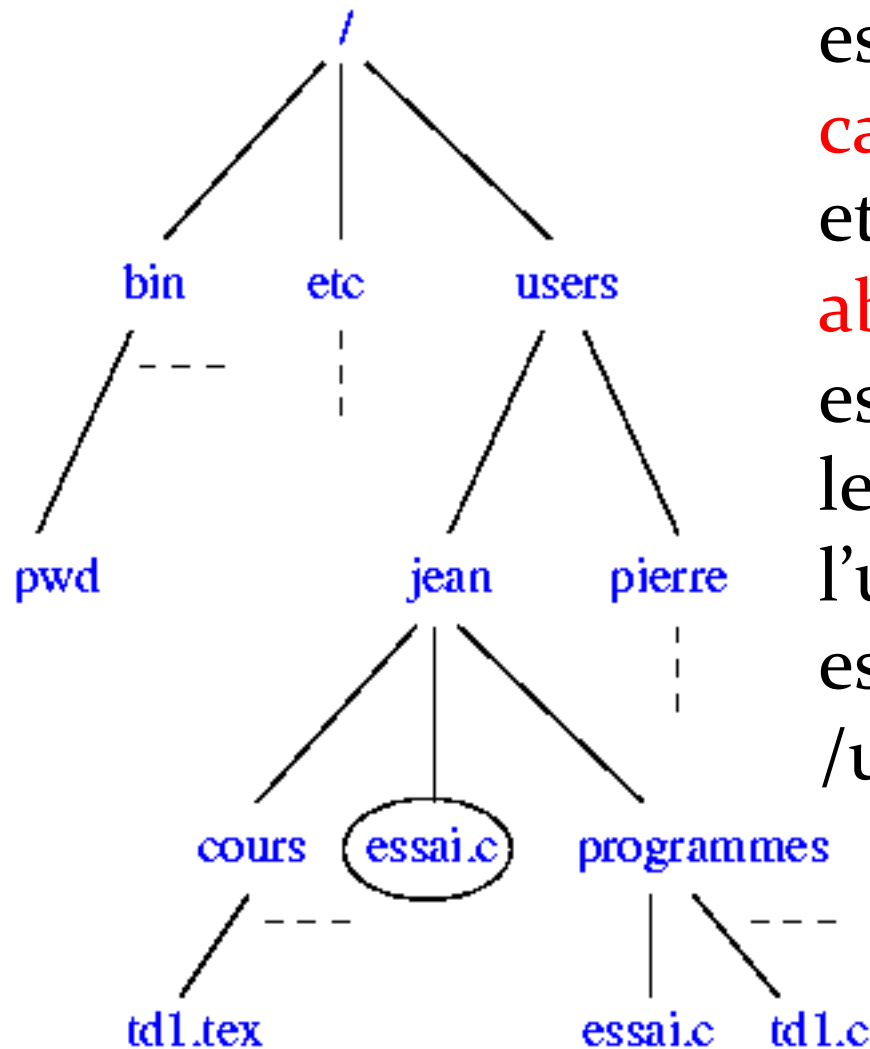
Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation



La racine de l'arbre est notée par le caractère / (slash) et le chemin d'accès absolu du fichier `essai.c` situé dans le répertoire de l'utilisateur `jean` est `/users/jean/essai.c`

Plan

Systèmes
d'exploitation

Fichiers

Processus

Compilation

Fichiers

Toute une série de commandes Unix gérant les fichiers et leur arborescence peuvent être exécutées dans le shell depuis le répertoire de travail (*working directory*) d'où a été lancé l'interpréteur. Ainsi :

ls	liste les fichiers du répertoire
rm	supprime des fichiers (<i>remove</i>)
mv	déplace un fichier (<i>move</i>)
pwd	affiche le répertoire de travail
cd	change de répertoire de travail

Plan

Systèmes
d'exploitation

Fichiers

Processus

Compilation

```
% cd
% pwd
/users/jean
% ls
Cours  essai.c  Programmes
% cd ../
% pwd
/users
% ls
jean   pierre   sarah
% cd
% rm essai.c
% ls
Cours          Programmes
```

année universitaire: 2016 -2017

Plan

Systèmes
d'exploitation

Fichiers

Processus

Compilation

Fichiers

Un fichier Unix a un propriétaire (le nom d'un utilisateur) et divers attributs comme sa taille, son nom, son chemin d'accès absolu, mais également des « droits d'accès » caractérisant la ou les possibilités pour différents utilisateurs de le lire, le modifier, ou de lancer son exécution.

Dans un programme, on pourra créer des fichiers, les « ouvrir » (avant de les lire ou de les modifier), ou les « fermer » (pour signaler au SE qu'on ne les utilisera plus).

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Processus

On appelle *processus* un programme informatique en cours d'exécution.

Le SE lorsqu'il est lancé met en route des processus permettant aux utilisateurs de se connecter (*login*). Il lance ensuite pour chaque utilisateur connecté un shell interpréteur de commande (accessible depuis la console), et l'utilisateur peut alors lui aussi lancer des programmes ou des logiciels qui créeront eux-mêmes de nouveaux processus susceptibles d'en créer d'autres.

Processus

Il existe de nombreuses commandes Unix permettant de manipuler les processus actifs sur l'ordinateur depuis la console ou un terminal.

```
ex:  $ ps -x
      PID      TT  STAT      TIME     COMMAND
      54120  ttys000      0:00.03  -bash
      54196  ttys000      0:00.00  cat
      54215  ttys000      0:00.00  ps -x
      $ kill -1 54196
```


Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Processus

Le Système d'Exploitation assure la gestion des processus :

- Le chargement des programmes de la mémoire de masse vers la mémoire de travail.
- Le partage du temps du processeur.
- Le partage de la mémoire.
- Les interruptions matérielles, et la communication entre processus par l'intermédiaire de l'envoi de signaux.

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Temps partagé

Dans un système multi-tâche, il y a un seul processeur (ou un petit nombre de processeurs) et de nombreux processus.

Chaque processus est exécuté tour à tour, pendant un intervalle de temps assez court (de l'ordre de 10 à 100 ms) jusqu'à ce que son tour revienne à nouveau. Cela donne l'illusion de la simultanéité et du parallélisme. On parle de **temps partagé** (*time sharing*).

C'est un programme du SE (l'ordonnanceur) qui détermine l'ordre dans lequel les processus sont exécutés.

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Mémoire partagée

Les adresses mémoires manipulées par les processus ne sont pas directement les adresses physiques, elles sont virtuelles. Un calcul d'adresse est effectué par le système pour passer des adresses virtuelles aux adresses physiques et réciproquement.

Cela permet de faire comme si chaque programme était seul en mémoire centrale et simplifie le travail de l'ordonnanceur et des programmeurs. Il y a néanmoins un léger coût (en temps) pour le changement de contexte.

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Environnement de travail

Le système peut aussi être vu comme un ensemble de logiciels qui accompagnent le matériel et sont installés en même temps que le système d'exploitation proprement dit. Mais on parle alors d' **environnement de travail**. Cela inclut

- Un environnement graphique (fenêtres, bureau, corbeille, gestionnaire de fichiers, etc.)
- Des services inter-applications (correction orthographique, couper/coller, annuaire, etc.)
- Des applications diverses : navigateur web, éditeur de texte, mail, environnement de programmation dédié, etc.

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Votre environnement de programmation en TP

Votre environnement de programmation en TP pourra être constitué simplement de :

- L'environnement graphique (fenêtres, bureau, corbeille, gestionnaire de fichiers, etc.) fourni par le SE.
- une fenêtre de terminal (fenêtre de lancement de vos commandes).
- un editeur de texte de programme (lancé depuis le terminal avec gedit par exemple)

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Compilation

La **compilation** est la traduction d'un texte de programme écrit dans un langage de programmation évolué, le **code source**, en un code binaire exécutable par la machine: le **code machine**.

Le langage C est un langage *compilé*. La traduction du code source en code machine s'effectue d'un seul tenant - contrairement aux langages *interprétés* (comme le noyau des commandes, le *shell*) où elle a lieu pas à pas, c-à-d ligne à ligne.

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Compilation

La compilation s'effectue en analysant et transformant plusieurs fois le fichier source. Les étapes de la compilation sont :

1. Prétraitement (préprocesseur)
2. Analyse lexicale
3. Analyse syntaxique
4. Analyse sémantique
5. Génération du code
6. Édition de liens

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Préprocesseur

Le préprocesseur du compilateur supprime les commentaires et exécute les instructions commençant par un dièse, c'est-à-dire les `#define` et les `#include`.

Le texte initial est transformé en un autre, par substitution textuelle.

```
ex: #define MAX 100
```

on remplace partout `MAX` par `100`

```
#include "fichier.h »
```

on substitue à cette ligne le texte du fichier

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Analyse lexicale

Elle identifie les **unités lexicales** (ou lexèmes) du texte. Les caractères blancs servent à les séparer, comme dans `int x`, mais ils sont sinon ignorés (`3 * x + 1`).

- Erreur lexicale : `int x = @;`

```
error: stray '@' in program
```

- Erreur détectée uniquement au moment de l'analyse sémantique : `intx = 0;`

```
error: 'intx' undeclared (first use in this function)
```

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Analyse syntaxique

Elle vérifie que la grammaire est conforme à celle du langage et construit un arbre syntaxique.

- exemple, le mot `else` apparaissant sans le `if` qui aurait du le précéder :

```
x = 10;
```

```
else x = 12;
```

```
error: expected expression  
before `else`
```

Plan

Systèmes
d'exploitation

Fichiers

Processus

Compilation

Analyse sémantique

Elle vérifie que le programme a bien un sens en listant les objets manipulés par le programme et en vérifiant que les actions du programme sur ces objets sont bien autorisées.

ex: variable `x` utilisée mais non déclarée

```
error: 'x' undeclared (first  
use in this function)
```

Remarque: l'inverse (variable déclarée mais non utilisée), génère un *warning* si on compile avec l'option `-Wall` de gcc.

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Génération de code

- La génération de code s'effectue en passant par un code en assembleur, avec optimisation et allocation des registres, suivi d'une traduction en **code objet**.
- Le code objet est un code qui n'est pas exécutable par la machine car il lui manque certaines parties, comme le code des fonctions de bibliothèques, d'une fonction utilisateur, ou celui de la fonction `main`.

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

On compilera un programme C sous Linux avec la commande gcc (compilateur GNU):

```
$ gcc -c fichier.c
```

→ crée un fichier de code objet fichier.o

On peut même créer un code objet avec un fichier main.o avec gcc -c main.c.

```
$ gcc main.c ou
```

```
$ gcc fichier.o main.c
```

→ crée un fichier exécutable a.out

```
$ gcc fichier.o main.c -o exec
```

→ crée un fichier exécutable exec.

année universitaire: 2016 -2017

Plan

Systèmes
d'exploitation

Fichiers

Processus

Compilation

Les options `-l` et `-L` indiqueront à l'éditeur de liens où chercher les fonctions de bibliothèques si elles ne sont pas trouvées dans les répertoires standards.

L'option `-Wall` permet d'avoir l'affichage de tous les messages d'avertissement (*warning*) de l'utilisateur.

L'option `-g` permet de créer un code qui sera interprétable par un *debugger*.

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

```
$ gcc -g -c fichier.c
```

```
$ gcc -g fichier.o main.c -o exec
```

```
$ gdb exec
```

permet de lancer le debogueur GNU gdb avec un executable exec (qui doit avoir été compilé avec l'option -g de gcc).

On peut aussi lancer gdb, puis charger un executable avec la commande file de gdb.

gdb une fois lancé, permet de lancer les commandes suivantes (entre autres!):

```
help file run list where quit
```

Plan

Systemes
d'exploitation

Fichiers

Processus

Compilation

Merci pour votre attention !

Des questions ?