

Introduction à l'informatique

Les processus

G. Santini, J.-C. Dubacq

IUT de Villetaneuse

S1 2016

Plan

1 Fichiers exécutable et Processus

Fichier binaire et fichier texte

Processus dans un système multitâches et mutli-utilisateurs

Gestion de la mémoire vive

Gestion de l'accès au CPU

Processus en ligne de commande

Plan

1 Fichiers exécutables et Processus

Fichier binaire et fichier texte

Processus dans un système multitâches et mutli-utilisateurs

Gestion de la mémoire vive

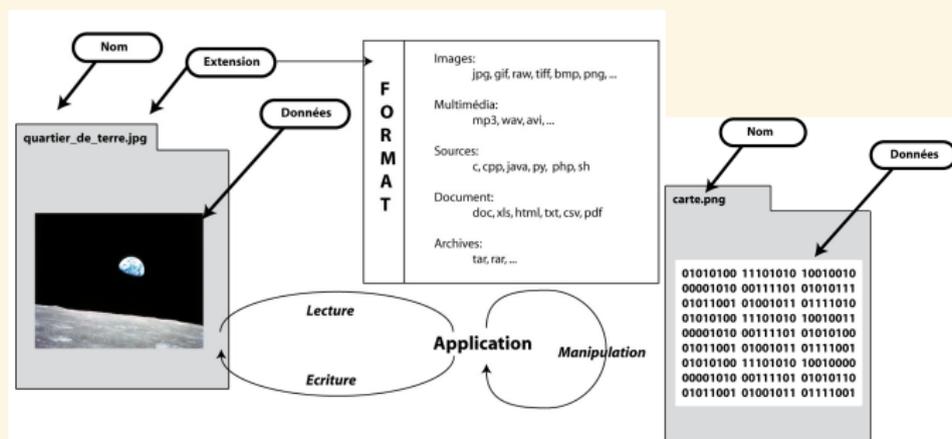
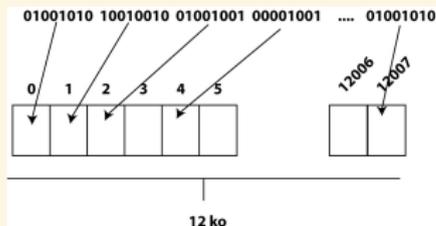
Gestion de l'accès au CPU

Processus en ligne de commande

Fichier binaire et fichier texte

Les données numériques

Tout fichier enregistré sur un support numérique est une suite d'octets.



Accès aux données

Lors de son utilisation un fichier est *lu* par un programme. Pour cela il doit décoder les informations binaires et les traiter.

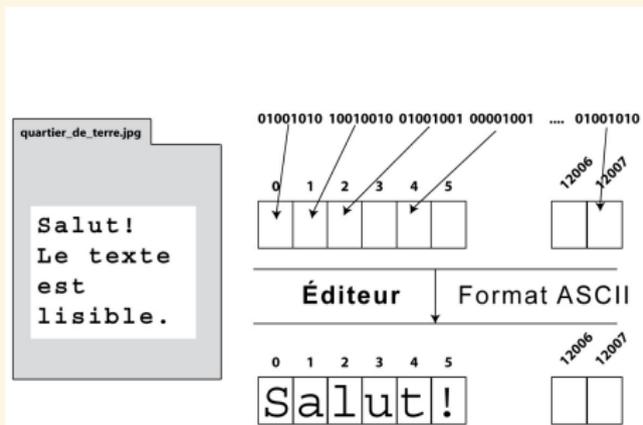
Fichier binaire et fichier texte

Deux grands types de fichiers : Binaire Vs Numérique

De façon générale un fichier binaire ne peut être "lu" que par un programme informatique, alors qu'un fichier texte peut être "lu" par être humain.

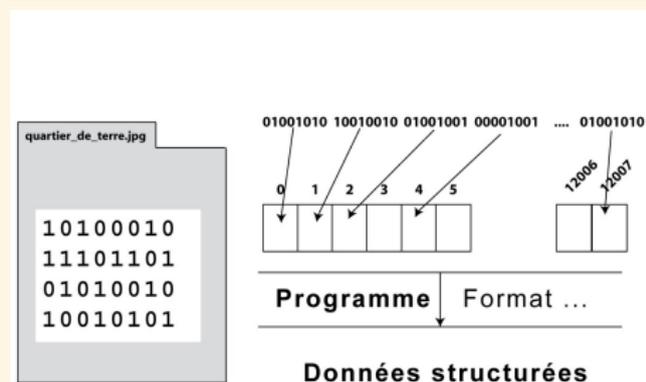
Les fichiers textes

C'est un fichier qui peut être "lu" par un éditeur de texte brut. Les données sont encodées comme une suite de caractères.



Les fichiers binaires

Ce n'est pas un fichier texte ... Il peut contenir des instructions machines, des données compressées, des données binaires brutes nécessitant un programme pour être lues.



Fichiers sources → Exécutable → Processus

Les sources : Une "recette de cuisine"

- ▶ Exprime un ensemble de tâches à réaliser pour accomplir le programme (le plat cuisiné).
- ▶ Utilise un langage de programmation.
- ▶ C'est un fichier texte.

dessine.c

```
(...)  
float r, x, y;  
r=3.0;  
x=0.0;  
y=7.1;  
cercle(0,0,,r)  
segment(0,0,x,y)
```

L'exécutable

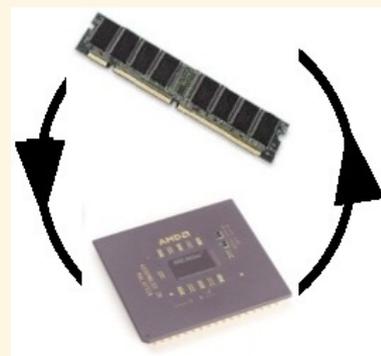
- ▶ Exprime les mêmes tâches dans un langage machine.
- ▶ Ce fichier ne fonctionne que sur des ordinateurs qui ont la même architecture.
- ▶ C'est un fichier binaire.

dessine

```
10100101 11101001  
10001001 00100101  
00101010 00100010  
01111011 10110101  
01000010 00110011  
00101101 11010100  
(...)
```

Les processus

- ▶ L'évaluation des instructions machines engendre des processus.
- ▶ Ces processus sont exécutés par le matériel.
- ▶ Les instructions machine doivent donc être adaptées au matériel.





Exercices

Préparation

1. Vérifiez que votre répertoire courant est bien `m1101`. Analysez l'affichage produit par la commande `ls` suivie des options `-lh`. Vous pourrez comparer les affichages obtenus par les commandes `ls -l` et `ls -lh` pour comprendre l'effet de l'option `-h`. Vous pourrez aussi rechercher cette information dans les pages de man.
2. Analysez l'arborescence créée lors de l'extraction des données de l'archive au moyen de la commande `ls`. Vous dessinerez cette arborescence.
3. Après vous être placé dans le répertoire créé lors de l'extraction de l'archive (`donnees`), quelle commande permet d'identifier le plus gros fichier (taille mémoire). Identifiez-le.
4. Quelles commandes vous permettent d'afficher le contenu des fichiers texte `command.txt` et `README`? Quels sont leurs contenus?
5. Analysez le résultat de l'évaluation des commandes suivantes :

```
file textes/README.txt
file textes/command.txt
file images/img_1175.jpg
```
6. Quelle est la fonction de la commande `file`? Parcourez les pages de manuel de cette commande.

Plan

1 Fichiers exécutable et Processus

Fichier binaire et fichier texte

Processus dans un système multitâches et mutli-utilisateurs

Gestion de la mémoire vive

Gestion de l'accès au CPU

Processus en ligne de commande

Identification des processus par le système d'exploitation

Système multi-utilisateur

- ▶ Plusieurs utilisateurs partagent les mêmes ressources matériel (RAM, CPU, disques, ...),
- ▶ Chaque utilisateur lance des processus liés à ses activités sur la machine et il utilise les résultats de ces processus.

Système multi-tâches

- ▶ Plusieurs programmes en cours d'exécution partagent les mêmes ressources matériel (mémoire vive, CPU, disques, ...). Ils peuvent provenir d'un seul ou de plusieurs utilisateurs,
- ▶ Chaque programmes lance des processus et il utilise les résultats de ces processus.

Il faut partager les ressources !!!

- ▶ Chaque programme doit être exécuté éventuellement "*en même temps*". Il faut donc gérer le partage des ressources de calcul (accès à la mémoire vive, au CPU),
- ▶ Chaque programme ou utilisateur doit pouvoir retrouver les résultats de ses calculs. Il faut donc pouvoir identifier qui a lancé les processus et qui doit récupérer les résultats.

La gestion des processus est réalisée par le système d'exploitation. C'est une de ses tâches principales. Pour cela il a besoin de pouvoir identifier chaque processus.

PID et PPID

PID - Process IDentifier

- ▶ C'est un numéro unique attribué à chaque processus lors de son lancement.
- ▶ Il permet d'identifier de façon unique chaque processus.
- ▶ La liste des processus en cours d'exécution est accessible en ligne de commande par les commandes `ps` et `top`.

PPID - Parent Process IDentifier

- ▶ Le premier processus lancé porte le numéro de PID 1. Les processus suivants sont des processus issus de ce processus parent.
- ▶ Chaque processus est lancé par un processus parent *via* l'appel système `fork`.
- ▶ Le PPID est le PID du processus Parent.

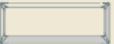
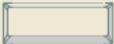
Utilités

- ▶ L'utilisateur peut suivre un processus, le suspendre temporairement, le relancer ou le tuer (interruption définitive).
- ▶ Le système s'en sert pour lui affecter des ressources matériel.



Exercices

Racourcis clavier et astuces en ligne de commande

7. Tapez les 2 caractères `s l` puis pressez la touche  (Tab). Que se passe-t-il ?
8. Tapez les 3 caractères `s l e` puis pressez la touche . Que se passe-t-il ?
9. À la suite de l'affichage précédent tapez la combinaison de touches  . Que se passe-t-il ?
10. Que fait la commande `man s l e e p` ? Que pouvez-vous dire de la commande `s l e e p` ?
11. Exécutez la commande `s l e e p 32000000`. Que se passe-t-il si vous tapez la combinaison de touches   ?
12. Quelle action produit la pression de la flèche  sur votre clavier ?
13. Quelle est l'action produite par la pression de la combinaison de touches   après avoir tapé quelques lettres ? Par la combinaison de touche   ?
14. Quelle est l'action produite en tapant `ls`   (le caractère  signifie la présence d'un espace) ?

Plan

1 Fichiers exécutable et Processus

Fichier binaire et fichier texte

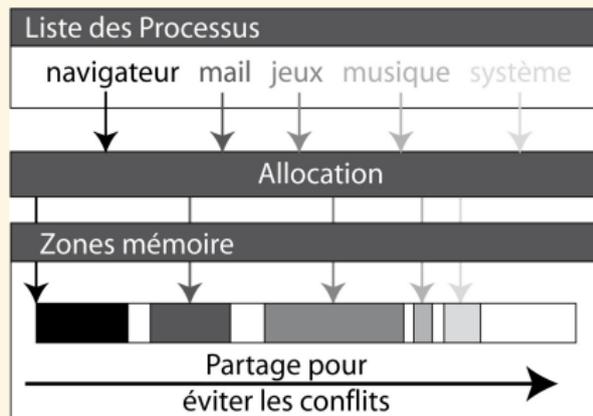
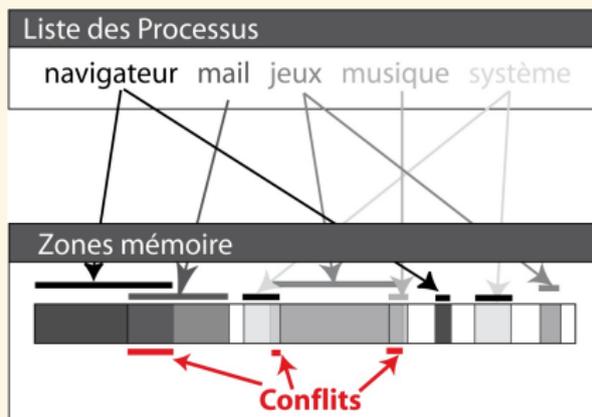
Processus dans un système multitâches et mutli-utilisateurs

Gestion de la mémoire vive

Gestion de l'accès au CPU

Processus en ligne de commande

Gestion de la mémoire vive



Chaque processus a besoin de mémoire

Pour stocker et travailler sur :

- ▶ les données,
- ▶ les instructions,
- ▶ les résultats.

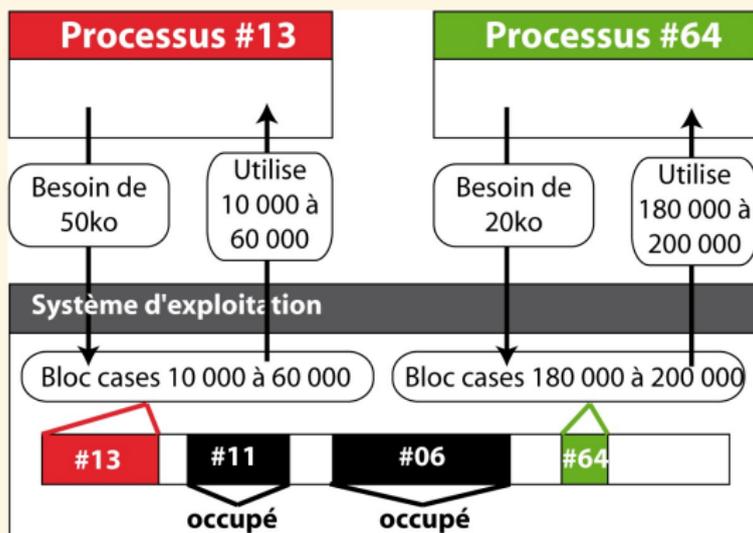
Il faut assurer l'intégrité des données !

Allocation de zone mémoire

L'allocation permet :

- ▶ d'attribuer à chaque processus un espace de travail en mémoire,
- ▶ le système contraint le programme à écrire dans sa zone mémoire et ainsi,
- ▶ évite qu'un programme modifie les données d'un autre programme.

Gestion de la mémoire vive



Principes généraux de l'allocation

- ▶ L'OS maintient une table des zones mémoires allouées à chaque processus. Ces zones sont réservées et ne peuvent être utilisées que par le processus parent.
- ▶ Lorsqu'il a besoin de mémoire, un processus demande à l'OS quelle zone il peut utiliser,
- ▶ L'OS lui attribue, en fonction de l'espace libre, un certain nombre de blocs mémoire.
- ▶ Les blocs mémoire attribués sont alors réservés.

Plan

1 Fichiers exécutable et Processus

Fichier binaire et fichier texte

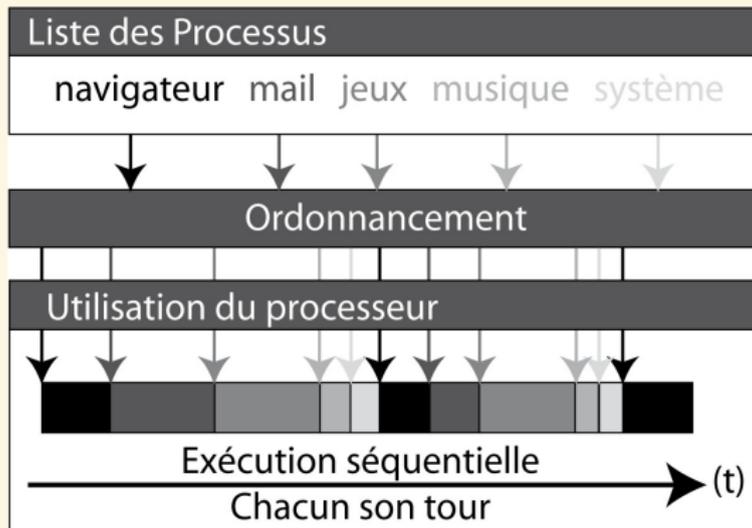
Processus dans un système multitâches et mutli-utilisateurs

Gestion de la mémoire vive

Gestion de l'accès au CPU

Processus en ligne de commande

Gestion de l'accès au CPU



Le planificateur gère le temps CPU attribué à chaque processus

- ▶ Le CPU ne traite qu'un seul processus à la fois,
- ▶ Le planificateur permet l'alternance d'accès au CPU en attribuant une priorité à chaque processus.
- ▶ L'illusion d'exécution simultanée de plusieurs processus est donnée par une alternance rapide d'attribution de temps de calcul à chaque processus.

Syntaxe pour ps

```
ps <-eu>
```

Description

- ▶ Affiche les processus en cours d'exécution.
- ▶ L'option <- e> indique que tous les processus doivent être affichés,
- ▶ L'option <- u> restreint l'affichage aux processus de l'utilisateur.

Exemple d'utilisation:

```
login@host:~$ ps -eu
Warning: bad ps syntax, perhaps a bogus '-?' See http://procps.sf.net
  USER  PID %CPU %MEM  VSZ   RSS  TTY  STAT  START  TIME  COMMAND
santini 5905  0.0  0.2 4824 1656 pts/1  Ss   09:27 0:00  -bash LC_ALL=fr_FR.UTF
santini 5962  0.0  0.1 3884  896 pts/1  R+   09:48 0:00  ps -eu MANPATH=/etc/jav
login@host:~$ █
```

Syntaxe pour top

top

Description

- ▶ Permet de suivre dynamiquement (temps réel) les ressources matériel utilisées par chaque processus.
- ▶ Ouvre un interface dans la ligne de commande qui peut être quittée en pressant la touche 
- ▶ Donne pour chaque processus en autres choses, le PID, le nom du propriétaire, la date de lancement du processus, les %CPU et %MEM utilisés.

Exemple d'utilisation:

```
Tasks: 85 total, 1 running, 84 sleeping, 0 stopped, 0 zombie
Cpu(s): 5.7%us, 0.0%sy, 0.0%ni, 93.6%id, 0.0%wa, 0.7%hi, 0.0%si, 0.0%st
Mem: 772068k total, 231864k used, 540204k free, 2412k buffers
Swap: 995992k total, 0k used, 995992k free, 161316k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5116	root	20	0	33832	22m	6576	S	5.7	3.0	0:19.49	X
5879	santini	20	0	16060	7344	6116	S	0.3	1.0	0:01.06	xfce4-netload-p
1	root	20	0	1664	568	496	S	0.0	0.1	0:02.95	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0

Plan

1 Fichiers exécutable et Processus

Fichier binaire et fichier texte

Processus dans un système multitâches et mutli-utilisateurs

Gestion de la mémoire vive

Gestion de l'accès au CPU

Processus en ligne de commande

Processus en ligne de commande

Occupation de la ligne de commande

- ▶ Lorsque l'on tape une commande, la ligne de commande est bloquée (plus de prompt) jusqu'à la fin de l'exécution.
- ▶ La ligne de commande est à nouveau disponible ensuite.

```
login@host:~$ sleep 20  
(il faut attendre 20 secondes avant  
l'apparition du nouveau prompt)
```

...

```
login@host:~$ █
```

```
login@host:~$ gedit  
(Il faut quitter l'application ou tuer  
le processus gedit pour avoir un nouveau  
prompt)
```

...

...

Libération de la ligne de commande

Deux façons possibles de lancer une instruction en tâche de fond :

Lancement en tâche de fond

- ▶ Les commandes qui prennent beaucoup de temps peuvent être lancées en tâche de fond pour libérer la ligne de commande du shell.
- ▶ Pour lancer directement la commande en tâche de fond il suffit de faire suivre la commande du caractère **&**. On retrouve immédiatement un nouveau prompt.

```
login@host:~$ gedit &  
login@host:~$ █
```

Relégation en tâche de fond

- ▶ Si une tâche déjà lancée occupe la ligne de commande, il est possible de suspendre son exécution en pressant la combinaison de touches **Ctrl** **Z**. La tâche est alors interrompue et on retrouve un nouveau prompt.
- ▶ Il est possible de relancer le processus en tâche de fond au moyen de la commande **bg**.

```
login@host:~$ gedit  
^Z  
[1]$+ Stopped gedit  
login@host:~$ bg  
[1]$+ gedit &  
login@host:~$ █
```



Exercices

Gestion des processus

Afin d'illustrer la gestion des processus nous allons utiliser la commande `sleep` pour simuler l'exécution de programmes dont l'exécution n'est pas immédiate. Pour se rappeler de son fonctionnement vous pouvez utiliser la commande `man`.

- Évaluez l'instruction `sleep 1000` puis tapez `Ctrl C`. Que se passe-t-il ?
- Évaluez l'instruction `sleep 1000 &` (n'oubliez pas le caractère `&`). Que se passe-t-il ?
- La commande `ps` permet d'afficher la liste de processus qui s'exécutent sur votre ordinateur. Un processus s'exécutant sous Linux est identifié par un numéro de processus, et par un propriétaire (celui qui a lancé le processus). Identifiez ces deux données lors de l'appel des commandes suivantes, donnez une explication à la différence des affichages (utilisez le `man` si nécessaire) :

```
ps  
ps -ef
```
- Quel est le numéro de processus associé à la commande `sleep 1000 &` ?



Exercices

Gestion des processus (suite)

11. La commande `kill` permet de « tuer » (supprimer) un processus. Sa syntaxe d'utilisation est la suivante : `kill PID` où PID (Process ID) doit être remplacé par le numéro du processus à supprimer.
12. Quelle commande permet de détruire le processus associé à la commande `sleep 1000 & ?`
13. Tapez la commande `gedit` dans le terminal. Quel est l'effet sur la ligne de commande ? Pouvez-vous saisir de nouvelles commandes ?
14. Après avoir lancé `gedit` (celui-ci étant en cours d'exécution), que se passe-t-il si on tape   dans le terminal qui a lancé `gedit` ? Quel est l'effet sur le programme `gedit` (utilisez `ps` pour suivre l'état des processus) ? Que se passe-t-il si vous tapez `bg` ?
15. Que fait la commande `top` ?
16. Exécutez la commande `ps -ef`. Examinez comment est construite la *forêt* de processus. Repérez comment sont agencés les processus qui gèrent vos terminaux entre eux.