









Introduction aux systèmes d'exploitation et à leur fonctionnement

Shell

Fabien Laguillaumie (fabien.laguillaumie@umontpellier.fr)





Objectifs de cette partie

Introduire les processus et les concepts de base pour les gérer

- Étudier les programmes qui peuvent s'exécuter (exécutables)
- Définir les processus
- Étudier les systèmes multi-tâches
- ► Introduire les concepts de base de la gestion des processus : les interruptions, l'état, l'ordonnanceur, les signaux ...

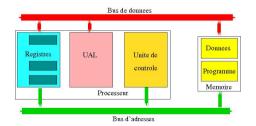




EXÉCUTABLES

Une première vue de la machine

- ▶ Pour comprendre l'exécution des programmes, on a besoin d'un schéma simple de la machine.
- ▶ Dans les architectures actuelles, il y a des blocs fonctionnels importants :
 - unité centrale (CPU) qui exécute automatiquement les instructions reçues
 - mémoire qui stocke les instructions et les données à manipuler
 - des canaux de communication, des périphériques, ...





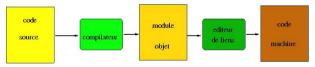


EXÉCUTABLES

Un programme exécutable

Il doit être directement compréhensible pour le CPU

- codé en binaire
- Il contient des instructions (en binaire) qui seront exécutées par le processeur
- Pour faire un programme exécutable, à partir de langage de programmation de haut niveau, on utilise la compilation



C'est toujours un code machine (binaire, compilé) qui s'exécute



Question : un script est-il un exécutable?



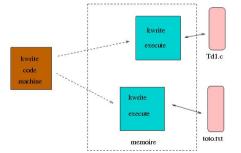
EXÉCUTABLES

Lancer un exécutable

- On peut lancer un exécutable à partir d'un shell (par exemple) en donnant le nom du fichier de l'exécutable (et des éventuels paramètres)
 - via le PATH ou par son chemin absolu (ex. : ./monexec)
 - ► Il doit avoir le droit d'accès pour l'exécution
- ► Un exécutable peut être lancé plusieurs fois

login@machine:~\$ kwrite &

login@machine:~\$ kwrite &







PROCESSUS

Une première définition :

- ► Un processus représente les activités d'un programme (d'une occurrence d'un programme) qui s'exécute
 - À un instant donné, un processus peut être caractérisé par un état.
 - Cet état comprend toutes les informations, valeurs qu'il faut savoir pour pouvoir continuer le processus
 - l'état du processeur
 - les registres, l'environnement du processus, les fichiers ouverts, ...





PROCESSUS

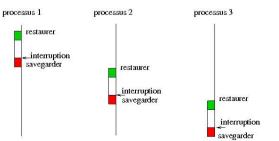
Exécution d'un processus

- Quand le processeur exécute un processus, il est amené à faire des pauses de temps en temps
 - Exemple 1:
 - Le processus veut lire des données du disque dur
 - ightharpoonup La lecture d'un secteur du disque est ~ 50.000 fois plus lente que la lecture d'une donnée en mémoire
 - Que faire pendant ce temps?
 - Exemple 2:
 - ▶ Un processus monopolise le processeur depuis un bon moment
 - Le système peut interrompre ce processus





- ▶ Pour simplifier, supposons un seul processeur
- Pour mieux organiser le temps du processeur, plusieurs processus partagent le CPU
 - le CPU est donné à un processus pour s'exécuter à un moment donné
 - quand l'exécution du processus est interrompue, un autre processus sera associé au CPU
 - chaque processus avance un peu, et le système donne la main à un autre pour avancer ...
- Globalement, on constate une quasi-parallélisme et une exécution indépendante des processus





Interruptions

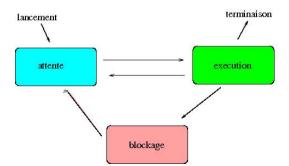
- Les interruptions sont primordiales pour assurer la bonne performance de toutes les exécutions.
- Qui doit interrompre une exécution et à quel moment?
- Exemples:
 - Le processus appelle une activité longue, extérieure (e.g., disque dur)
 - L'utilisateur intervient (e.g., frappe une commande au clavier)
 - L'exécution d'un processus est trop longue (e.g., boucle infinie)
- L'interruption peut alors venir
 - d'un matériel (disque dur, clavier, etc.)
 - du code du programme
 - du système (qui ne veut pas qu'un processus monopolise le CPU)





L'exécution des processus

- ▶ Un automate peut illustrer les différents grands états d'un processus
- ► Le processus peut être
 - en attente
 - actif
 - bloqué

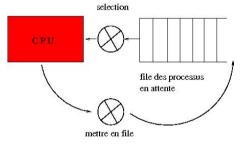






Ordonnancement

Le schéma général de la gestion de l'exécution peut être résumé comme suit



- ▶ Il existe un ordonnanceur (scheduler) pour gérer les processus
- Des politiques différentes peuvent être envisagée pour gérer les files d'attente
- **Exemples**:
- réaliser un tourniquet (fair play) sélection basée sur des priorités



Gestion dans Linux

- L'ordonnanceur de Linux associe une priorité à chaque processus
 - Dans un premier temps, cette priorité est maximale pour le processus lancé
- Si le processus occupe fortement le CPU, le système baisse sa priorité (et inversement)
- L'utilisateur peut lancer son processus avec une priorité (*niceness*) baissée avec la commande

login@machine:~\$ nice -n 10 monexecutable

- la priorité est diminuée de 10
- L'utilisateur peut changer la priorité de son processus avec la commande renice
- ▶ Il ne peut donner qu'une priorité moins importante à son processus...
- ▶ Seul root peut donner des valeurs "négatives" pour augmenter les priorités





HIÉRARCHIE DES PROCESSUS

Quand un processus est lancé, on connait le processus qui l'a lancé

- On peut dire qu'un processus a un père : le processus qui l'a lancé Ex. : si un exécutable est lancé à partir d'un shell, ce dernier est le père du processus
- Le premier processus est le processus init quand le système est lancé
- ▶ À partir de cet antécédent commun, on peut envisager un *arbre* des processus, visualisable via

login@machine:~\$ pstree -C age -nup





HIÉRARCHIE DES PROCESSUS

Table des processus

- Pour administrer les processus, le système possède une table
- ▶ Dans cette table, chaque processus possède un numéro : PID
 - Le premier processus init possède l'indice 1
- Éléments dans la table (liste non exhaustive)
 - ► PID
 - utilisateur (owner)
 - priorité
 - variables d'environnement
 - ► indice du père (PPID)
 - pointeurs pour l'exécutions ...

Pour le shell : UID, GID, BASHPID, PPID

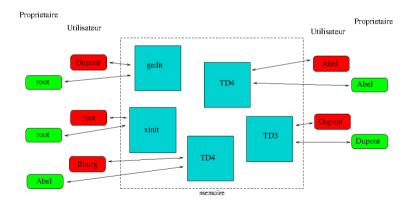




HIÉRARCHIE DES PROCESSUS

Exécution de processus

▶ Un exemple







VISUALISATION DES UTILISATEURS ET DES ACTIVITÉS

Qui est connecté?

La commande w permet de voir les utilisateurs connectés

```
laguillaumie@fabuntu:~$ W

10:39:58 up 1:42, 2 users, load average: 1,33, 1,33, 0,98

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

laguilla tty2 tty2 08:57 1:42m 0.01s 0.01s /usr/libexec/gnome-session-binary --systemd --session=ubuntu fauxuser pts/2 - 10:39 22.00s 0.08s 0.03s -bash

laguillaumie@fabuntu:~$
```

▶ Vous pouvez aussi regarder la commande who





QUELS PROCESSUS UTILISENT LE CPU?

La commande top (htop) permet de voir les activités actuelles

ıπ					fauxus	er@fabu	ntu	: ~		Q =	-	0	×
	0:42:29 up												
	346 total,									zombie			
	11,7 us,											sτ	
	: 31964 ,												
nrp swar). 976,	0 ()tat,	, 970	, o rree	'	0,0	useu.	2530	, Z avatt	нен		
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAI	ND	
	laguill+	20	0	5533544		338412	S	67.0	3,0	21:14.35			_
	laguill+	20	0	2499588	156220	110516	R	25,1	0,5	5:09.75	Web Co	onten	t
2211	laguill+	20	0	6837484	444800	240580	S	17,8	1,4	5:03.77	gnome	-shel	ι
248	root	-51	0	0	0	0	S	2,6	0,0	0:20.23	irq/5	1-DEL	L+
5431	laguill+	20	0	3209388	547764	153448	S	1,7	1,7	3:39.62			
9475	root	20	0	0	0	0	I	1,7	0,0	0:00.73	kworke	er/6:	1+
50	root	20	0	0	0	0	S	1,0	0,0	0:15.16	ksoft	irqd/	5
5352	laguill+	20	0	3127260	316104	169224	S	1,0	1,0	0:42.92	Web Co	onten	t
993	avahi	20	0	9960	5288	4328	S	0,7	0,0	0:07.21	avahi.	-daem	on
5270	laguill+	20	0	2851728	272880	120624	S	0,7	0,8	1:10.18	Web Co	onten	t
5448	laguill+	20	0	2915852	286472	141348	S	0,7	0,9	0:45.94	Web Co	onten	t
9491	fauxuser	20	0	15080	5032	4052	R	0,7	0,0	0:00.17	top		
12	root	20	0	0	0	0		0,3	0,0	0:00.07	ksoft	irqd/	Э
	root	20	0	0	0		I	0,3	0,0	0:03.58			
499	root	-51	0	0	0	0	S	0,3	0,0				
	root	0	-20	0	0	_	D	0,3	0,0				
5722	laguill+	20	0	2866632	278680	101808	S	0,3	0,9	0:39.75	WebEx1	tensi	+





L'INTENSITÉ DE L'UTILISATION

► Elle peut être illustrée par tload

F				1	auxuser(@fabuntu	:~			Q	≡	-	o <u>(8</u>)
1,67, 1,4	1, 1,26								**				
*									***				

**	*		**						*****	*		*	**
***	**	*	***						*****	*		*	* **
****	*** *	***	*** *						*****	*		*	* ***
*****	**** **	****	*****						*****	**		***	****
******	******	*****	***** *						***** *	**	**	****	*****
******	******	******	***** **	*			*		******	****	***	****	*****
******	*****	*****	*****	**			*		******	****	* ***	****	*****
*****	******	*****	*****	**			**		******	*****	* ***	****	*****
******	******	******	******	***		**	**	*	*******	*****	****	****	*****
******	******	******	******	***	*	**	***	***	*******	*****	****	****	*****
******	******	******	******	***	*** *	****	***	****	******	*****	****	****	*****
******	******	******	******	****	*** * :	* *****	****	****	******	*****	****	****	*****
******	******	******	******	*****	*****	******	****	****	*******	*****	****	****	*****
									-******	*****	*====		
******	*****	*****	******	*****	*****	******	****	*****	******	*****	****	****	*****
******	*****	*****	******	*****	*****	******	****	*****	******	*****	****	****	*****
******	*****	******	******	*****	*****	******	****	*****	******	*****	****	****	*****
******	******	******	******	*****	*****	******	****	*****	******	*****	****	****	*****
******	*****	******	******	*****	*****	******	****	*****	******	*****	****	****	*****
******	*****	*****	******	*****	*****	******	****	*****	******	*****	****	****	*****
******	******	*****	******	*****	*****	******	****	*****	******	*****	****	****	*****
******	******	*****	******	*****	*****	******	****	*****	******	*****	****	****	*****
*****	******	*****	******	*****	*****	*****	****	*****	******	*****	****	****	*****
******	*****	*****	******	*****	*****	*****	****	****	*****	*****	****	****	*****
******	******	*****	******	*****	*****	*****	****	****	******	****	****	****	*****
******	******	*****	******	*****	*****	*****	****	****	******	****	****	****	*****
******	******	******	******	*****	*****	*****	****	*****	******	*****	****	****	*****
*****	******	*****	******	*****	*****	*****	****	****	******	*****	****	****	*****
*****	*****	*****	*****	*****	*****	*****	****	****	*****	****	****	****	*****
*****	******	******	******	*****	*****	*****	****	****	******	*****	****	****	*****





VISUALISATION DES PROCESSUS

La commande ps

- Elle permet de visualiser des processus enregistrés dans la table des processus
- Quelques options:
 - Processus lancés depuis le terminal (le shell)
 - Visualisation de tous les processus
 - Visualisation de toutes les informations
 - Naturellement, les options peuvent être couplées

Pour trouver les processus lancés par user login@machine:~\$ ps -ef | grep user

Lire le manuel pour la distinction POSIX, BSD, GNU





login@machine:~\$ ps

login@machine:~\$ ps -e login@machine:~\$ ps -a

login@machine:~\$ ps -f

login@machine:~\$ ps -ef

VISUALISATION DES PROCESSUS

La commande ps

				١	laguilla	aumie@fabuntu:	~ Q = 8
laguilla	umie@fabur	ntu:~\$ p	s				
PID	TTY	TIME	C	MD			
10810	pts/2 6	00:00:00	b	ash			
10830		00:00:00					
laguilla	umte@fabur	ntu:~\$ p:	S	-ef			
UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0		08:57			/sbin/init splash
root	2	0		08:57		00:00:00	
root	3	2		08:57		00:00:00	
root	4	2		08:57		00:00:00	
root	6	2		08:57		00:00:00	
root	9	2		08:57		00:00:00	
root	10	2		08:57			[rcu_tasks_rude_]
root	11	2		08:57		00:00:00	
root	12	2	0	08:57	?	00:00:00	
root	13	2	0	08:57	?	00:00:05	[rcu_sched]
root	14	2	0	08:57	?	00:00:00	
root	15	2		08:57		00:00:00	
root	16	2	0	08:57	?	00:00:00	
root	17	2	0	08:57	?	00:00:00	
root	18	2	0	08:57	?	00:00:00	[idle_inject/1]
root	19	2		08:57			[migration/1]
root	20	2		08:57			[ksoftirqd/1]
root	22	2		08:57			[kworker/1:0H-events_highpri]
root	23	2	0	08:57	?		[cpuhp/2]
root	24	2		08:57			[idle_inject/2]
root	25	2	0	08:57	?	00:00:00	
root	26	2		08:57			[ksoftirqd/2]
root	28	2		08:57			[kworker/2:0H-events_highpri]
root	29	2	0	08:57	?		[cpuhp/3]
root	30	2	0	08:57	?		[idle_inject/3]
root	31	2	0	08:57	?	00:00:00	[migration/3]
root	32	2	0	08:57	?	00:00:00	[ksoftirad/3]





AVANT PLAN ET ARRIÈRE PLAN

- Quand on lance un processus, en général, il a
 - l'entrée standard (clavier)
 - la sortie standard (fenêtre)
- ▶ Il peut communiquer en utilisant ces dispositifs : il est en avant plan

```
laguillaumie@fabuntu:~

laguillaumie@fabuntu:~

laguillaumie@fabuntu:~

cat

une ligne

une ligne

et une deuxième

et une deuxième

...

...

^C

laguillaumie@fabuntu:~

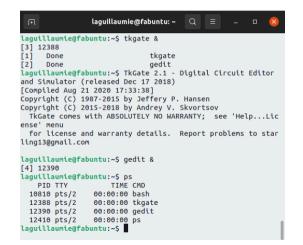
...
```





AVANT PLAN ET ARRIÈRE PLAN

- On peut lancer un processus privé du clavier et de la fenêtre du shell : login@machine:~\$ executable &
- ▶ Un processus en arrière plan n'a ni le clavier ni le terminal du shell







Avant plan et arrière plan

Quelques commandes utiles :

▶ Pour envoyer un processus en arrière plan

```
Ctrl + Z (pour stopper)
login@machine:~$ bg n°
```

Pour remettre un processus en avant plan :

```
login@machine:~$ fg n°
```

Exemple :

```
laguillaumie@fabuntu: ~
laquillaumie@fabuntu:~$ xclock &
[1] 8825
laguillaumie@fabuntu:~$ gedit
۸z
[2]+ Stopped
                              aedit
laquillaumie@fabuntu:~S iobs
[1]- Running
                              xclock &
[2]+ Stopped
                              aedit
laquillaumie@fabuntu:~S bg 2
[2]+ gedit &
laquillaumie@fabuntu:~$ jobs
[1]- Running
                              xclock &
[2]+ Running
                              gedit &
laquillaumie@fabuntu:~S fg 1
xclock
laguillaumie@fabuntu:~S
```

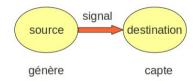




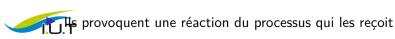
SIGNAUX

Les signaux réalisent des « événements » pour influencer l'exécution des processus

- ▶ Ils permettent de communiquer entre les processus
- Un signal est généré et envoyé par un processus et il est reçu par un autre processus



Les signaux correspondent à des nombres entiers





MONTPELLIER-SETE 2

SIGNAUX

Quelques signaux

- Souvent, on utilise des noms symboliques pour les référencer
- Exemples :
 - SIGALRM alarme, pour réveiller l'horloge du processus
 - SIGILL pour indiquer une instruction illégale
 - SIGINT interruption de l'exécution
 - SIGKILL pour tuer le processus
 - SIGSTOP pour stopper le processus
 - **▶** ...
- L'envoi des signaux est programmable (il y a des appels système pour le faire)
- La réaction du récepteur est aussi programmable





SIGNAUX

Deux exemples connus, en utilisant le shell

- ▶ Pour stopper le processus en avant plan : Ctrl + Z
- ▶ Pour tuer le processus en avant plan : Ctrl + C
- ► Comment ça marche?

