

Un premier approche

Les segments d'un processus

Exercice 1. Proposer un schéma d'un programme qui affiche la suite des ses sections (code, données, bss, pile,tas) ordonnées selon la position dans l'espace d'adressage du programme. Le programme affichera aussi la direction d'agrandissement de la pile et du tas dans l'espace d'adressage du programme. On pourra se servir de la fonction

```
#include <unistd.h>
void * sbrk(ptrdiff_t incr);
```

en sachant que l'appel de `sbrk(0)` retournera le premier adresse libre du tas.

Exercice 2. Implémenter le programme de l'exercice précédent.

Exercice 3. Dans le manuel de `sbrk` on lit :

CONFORMITÉ

BSD 4.3

`brk` et `sbrk` ne sont pas définis dans le C Standard, et sont volontairement exclus des standards POSIX.1 (paragraphes B.1.1.1.3 et B.8.3.3).

Expliquer la raison pour laquelle ces fonctions sont exclues des standards POSIX.1

Ordonnancement

Exercice 4. Dans le système UNIX, un processus mis en attente d'une opération d'entrée à partir d'un terminal est plus prioritaire qu'un processus mis en attente d'une opération de sortie vers ce terminal. Expliquer pourquoi.

Exercice 5. Une commutation de contexte prend c unité de temps et la durée moyenne de la phase de calcul d'un processus est de p unités de temps. Calculer le rendement de l'unité centrale en fonction de c, p , et de la valeur q du quantum, lorsqu'on utilise un tourniquet.

Exercice 6. Calculer et comparer les assignations produites par les algorithmes FIFO, PCTE, tourniquet (le quantum vaut 1), PCTER dans l'exemple suivant (ici τ_i dénote le temps d'exécution du processus P_i , et t_i son temps d'entrée dans la file d'attente) :

	τ_i	t_i
P_1	7	0
P_2	4	0
P_3	6	1
P_4	1	1
P_5	2	1
P_6	4	2
P_7	1	2

Exercice 7. Considérons la file d'attente suivante :

	τ_i	t_i
P_1	2	0
P_2	3	0
P_3	5	1
P_4	x	1

Donner, en fonction de x , l'assignation (c.-à-d. l'ordonnancement) sans réquisition, qui minimise le temps de traitement moyen.

Exercice 8. Écrire des fonctions qui simulent les algorithmes FIFO, PCTE, tourniquet, et PCTER.

Chaque fonction prendra en entrée une suite des processus, chaque processus P_i possédant un temps τ_i d'exécution et un temps t_i d'entrée dans la file d'attente.

On pourra se servir des implémentations des files d'attente qu'on peut trouver dans la page web du cours.

Exercice 9. Modifier les fonctions de l'exercice précédent pour écrire un programme qui prend en entrée la représentation d'une file d'attente et qui calcule le temps de traitement moyen selon les algorithmes FIFO, PCTE, tourniquet, et PCTER. Pour les derniers, on pourra passer la valeur du quantum en paramètre.

Exercice 10. Modifier le programme précédent pour prendre en compte la valeur c du « overhead » associée à la commutation de contexte.

Utiliser ce programme pour produire des valeurs expérimentales en fonction de la valeur q du quantum et de la valeur c du overhead.