



Examen du 14 janvier 2008 (1h)

RS : Réseaux et Systèmes
Deuxième année



La notation tiendra compte de la validité des réponses, mais aussi de la présentation et de la clarté de la rédaction.

Documents interdits, à l'exception d'une feuille A4 à rendre avec votre copie.

Questions de cours (4pt)

- ▷ **Question 1.** Définissez les fonctions principales des systèmes d'exploitation.
- ▷ **Question 2.** Que font `bash` et `gcc`? Quels sont les points communs et les différences entre leurs approches?
- ▷ **Question 3.** À quoi servent les appels systèmes `fork` et `exec`?
- ▷ **Question 4.** Qu'est ce que la pile d'un processus? Qu'est ce que le tas?
- ▷ **Question 5.** Qu'est ce qu'une condition de compétition? Dans quelles conditions cela se produit-il?
- ▷ **Question 6.** Comment prévenir les interblocages? Donnez et expliquez trois méthodes.
- ▷ **Question 7.** Qu'est ce qu'un processus zombie? Dans quelles conditions cela se produit-il?
- ▷ **Question 8.** Qu'est ce qu'une lecture tronquée? Dans quelles conditions cela se produit-il?

Exercices

Exercice 1 : Faire attendre un processus (1pt)

- ▷ **Question 9.** Écrivez une fonction nommée `void somnole(int n)` Elle bloque le processus courant le nombre de secondes indiqués en paramètre et affiche à chaque seconde le temps restant à dormir.

Exercice 2 : Traitement des signaux (1,5pt)

- ▷ **Question 10.** Que fait le programme 1 présenté plus bas?
- ▷ **Question 11.** Modifiez ce programme de façon à ce qu'il survive à un `<CTRL-C>` (et à un seul).

Exercice 3 : Clonage de processus (1,5pt)

- ▷ **Question 12.** Combien de lignes «plop!» impriment les programme 2 et 3 présentés plus bas?

```
Programme 1
1 #include "csapp.h"
2 void handler(int sig) {
3     printf("ctrl-z pressé !\n");
4 }
5 int main() {
6     Signal(SIGTSTP, handler);
7     pause ();
8     exit(0);
9 }
```

```
Programme 2
1 int main() {
2     int i;
3
4     for (i = 0; i < 3; i++){
5         printf("plop!\n");
6         fork();
7     }
8     printf("plop!\n");
9     exit(0);
10 }
```

```
Programme 3
1 void doit() {
2     fork();
3     fork();
4     printf("plop\n");
5     fork();
6 }
7 int main() {
8     doit();
9     printf("plop\n");
10    exit(0);
11 }
```

Exercice 4 : Édition de liens et symboles multiples (total : 2pt)

Pour chacun des cas ci-après, indiquez si la compilation et l'édition de lien des deux modules sont possibles. Si c'est impossible, indiquez pourquoi. Si les deux sont possibles, indiquez le résultat.

Dans tous les cas, argumentez vos affirmations; on ne demande pas la valeur des pointeurs affichés, mais ce vers quoi ils pointent.

▷ Question 1. (0,25 pt)

```

Module A
1 void proc(void) {
2   /* Quelques chose */
3 }
    
```

```

Module B
1 void proc(void) {
2   /* Quelques chose */
3 }
    
```

▷ Question 2. (0,25 pt)

```

Module A
1 int proc(void) {
2   return 42;
3 }
    
```

```

Module B
1 int proc(void);
2 int toto(void) {
3   return proc();
4 }
    
```

▷ Question 3. (0,5 pt)

```

Module A
1 int x;
2 int y;
3 int carre(int i) {
4   x = i;
5   y = i * i;
6   return y;
7 }
    
```

```

Module B
1 double x ;
2 int inv(int i) {
3   x = 1.0 / (double) i;
4   return x;
5 }
    
```

▷ Question 4. (1pt)

```

Module A
1 void p2(void);
2 int main() {
3   p2();
4   return 0;
5 }
    
```

```

Module B
1 #include <stdio.h>
2 char main;
3 void p2() {
4   printf("0x%x\n", main);
5 }
    
```