



# Examen du 15 juin 2006 (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.** Décrivez les deux types d'interfaces des systèmes d'exploitation courantes.
- ▷ **Question 2.** Que font `bash` et `gcc`? Quels sont les points communs et les différences entre leurs approches?
- ▷ **Question 3.** À quoi sert l'appel système `dup`?
- ▷ **Question 4.** Que font les commandes `jobs`, `fg` et `bg`?
- ▷ **Question 5.** Qu'est ce qu'un interblocage? Dans quelles conditions cela se produit-il?
- ▷ **Question 6.** Qu'est ce qu'une famine en informatique? Dans quelles conditions cela se produit-il?
- ▷ **Question 7.** Qu'est ce qu'une lecture tronquée? Dans quelles conditions cela se produit-il?
- ▷ **Question 8.** Définissez symboles forts et faible et donnez des exemples.

## 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 (2pt)

- ▷ **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 (1pt)

- ▷ **Question 12.** Combien de lignes «plop!» imprime le programme 2 présenté plus bas?

### Exercice 4 : Relecture de code (2pt)

- ▷ **Question 13.** Listez les (nombreuses) erreurs du programme 3 présenté plus bas, et corrigez-les.

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

```
Programme 2
int main() {
    int i;

    for (i = 0; i < 3; i++){
        printf("plop!\n");
        fork();
    }
    printf("plop!\n");
    exit(0);
}
```

```
Programme 3
int main(int argc, char *argv[ ]) {
    int i;
    pid_t pid;
    /* Creation de NBENFANTS enfants */
    for (i=0; i<NBENFANTS ; i++){
        pid = fork();
        switch(pid){
            case -1:
                perror(NULL);
                exit(EXIT_FAILURE);
            case 0:
                printf("Père: prêt\n");
                break;
            default:
                printf("Fils: clonage ok\n");
                break;
        }
    }
    /* un calcul long pour le père */
}
```