

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

*La notation tiendra compte de la présentation et de la clarté de la rédaction.*

*Le barème, approximatif, est donné sur 5 points puisque l'épreuve dure 30mn ( $\frac{5}{20} = 0,25 = \frac{30m}{2h}$ ).*

### ★ Questions de cours.(2pts)

- ▷ **Question 1:** À quoi servent respectivement les appels système `fork`, et `pipe` ?
- ▷ **Question 2:** Qu'est ce que la pile d'un processus ? Qu'est ce que le tas ?
- ▷ **Question 3:** Donnez deux signaux existant sous Unix. Dans quelles conditions sont-ils émis ? Quel est leur effet par défaut sur le processus les recevant ?
- ▷ **Question 4:** Qu'est ce qu'une lecture tronquée ? Dans quelles conditions cela se produit-il ?

### ★ Exercice 1: Clonage de processus (1pt) Considérez les programmes ci-dessous.

- ▷ **Question 1:** Combien de lignes «plop!» impriment-ils ?
- ▷ **Question 2:** Dessinez le schéma de filiation entre processus obtenu.

```

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

```

Programme 2
1 int main() {
2     if (fork() && fork())
3         printf("plop!\n");
4     printf("plop!\n");
5     exit(0);
6 }
    
```

### ★ Exercice 2: Édition de liens et symboles multiples (2pts).

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 du code (affichage, valeurs de retour, etc). On ne demande pas la valeur numérique des pointeurs, mais ce vers quoi ils pointent. Dans tous les cas, argumentez vos affirmations.

- ▷ **Question 1:** (0,25 pt)

```

fichier.1.c
1 int proc(void) {
2     return 42;
3 }
    
```

```

fichier.2.c
1 int proc(void) {
2     return 24;
3 }
    
```

```

fichier.3.c
1 int main(int argc, char**argv) {
2     printf ("%d", proc());
3 }
    
```

- ▷ **Question 2:** (0,25 pt)

```

fichier.1.c
1 int proc(void) {
2     return 42;
3 }
    
```

```

fichier.2.c
1 int proc(void);
2 int main(void) {
3     printf ("%d", proc());
4 }
    
```

- ▷ **Question 3:** (0,5 pt)

```

fichier.1.c
1 int x;
2 int y;
3 int suite_carre(int i) {
4     x += i;
5     y += i * i;
6     return y;
7 }
    
```

```

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

```

fichier.3.c
1 int x = 1;
2 int y = 2;
3 int main(int argc, char**argv) {
4     printf ("%f", inv(2));
5     printf ("%d", suite_carre(2));
6 }
    
```

- ▷ **Question 4:** (1pt)

```

fichier.1.c
1 void p2(void);
2 int main() {
3     p2();
4     return 0;
5 }
    
```

```

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