

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 (2pts)

- ▷ **Question 1:** Comment prévenir les interblocages ? Donnez et expliquez trois méthodes.
- ▷ **Question 2:** Qu'est ce qu'un processus zombie ? Dans quelles conditions cela se produit-il ?
- ▷ **Question 3:** Qu'est ce que POSIX ? Pourquoi cela existe-t-il ?
- ▷ **Question 4:** Définissez symboles forts et faibles et donnez des exemples.

## Exercice 1 : Clonage de processus (3pts)

- ▷ **Question 5:** Discutez les différents affichages possibles des programmes ci-dessous (avec schémas).

<pre style="margin: 0;"> 1 void main(int argc, char*argv[]) { 2   int i; 3   for (i=0; i&lt;n; i++) { 4     if (fork() == 0) { 5       exit(0); 6     } 7   } 8 }</pre> <p style="text-align: center; margin-top: 5px;">Programme 1</p>	<pre style="margin: 0;"> 1 void main(int argc, char*argv[]) { 2   int i; 3   for (i=0; i&lt;n; i++) { 4     if (fork() == 0) { 5       exit(0); 6     } else { 7       wait(NULL); 8     } 9   } 10 }</pre> <p style="text-align: center; margin-top: 5px;">Programme 2</p>	<pre style="margin: 0;"> 1 pid_t plop() { 2   fork(); 3   printf("plop\n"); 4   return fork(); 5 } 6 7 int main(){ 8   if(plop()) 9     if(plop()) 10      printf("yes\n"); 11 }</pre> <p style="text-align: center; margin-top: 5px;">Programme 3</p>
---	---	--

## Exercice 2 : Processus et tubes. (3pts)

Considérez le programme suivant. Il crée une série de processus ouvrant des tubes entre eux.

- ▷ **Question 6:** Dessinez l'organisation des processus et tubes telle qu'elle est en ligne 27. Il ne vous est pas demandé de dessiner l'historique menant à cette situation (comme à l'exercice précédent), mais bien la situation à l'instant où la ligne 27 est exécutée (en supposant que tous les processus exécutent cette ligne en même temps). Pensez à indiquer quel processus est créé à quel ligne sur votre schéma.
- ▷ **Question 7:** Décrivez en une phrase ce que font les processus créés par ce programme.

```

7 int main(int argc, char *argv[ ]) {
8   int tubA[2], tubB[2], tubC[2], tubD[2];
9   pipe(tubA); pipe(tubB); pipe(tubC); pipe (tubD);
10  int n1=0,n2=0;
11  close(0); close(1);
12
13  if ((n1=fork())) {
14    dup2(tubA[0],0);    dup2(tubB[1],1);
15  } else if ((n2=fork())) {
16    dup2(tubB[0],0);    dup2(tubC[1],1);
17  } else {
18    dup2(tubC[0],0);    dup2(tubA[1],1);
19    printf("A");
20    fflush(stdout);
21  }
22  close(tubA[0]); close(tubA[1]);
23  close(tubB[0]); close(tubB[1]);
24  close(tubC[0]); close(tubC[1]);
25  close(tubD[0]); close(tubD[1]);
26
27  /* Dessinez l'état des lieux à ce point */
28
29  char c;
30  while ((read(0,&c,1)>0) { /* Cette boucle ne s'arrête jamais */
31    write(1,&c,1);
32  }
33 }
```

### Exercice 3 : Feu de Foyer (2pts)

Suite à une expérimentation malheureuse, un début d'incendie s'est déclaré au club robotique. Immédiatement, tous les étudiants présents sur place et au foyer se précipitent à la cafet pour remplir des carafes d'eau avant de les jeter sur les flammes. Un seul robinet, des couloirs exigus et des dizaines d'étudiants courant en tout sens pour porter de l'eau vers les flammes : le chaos est indescriptible.

▷ **Question 8:** Proposez une solution à base de sémaphore(s) pour éviter que les étudiants se bousculent devant le point d'eau. De quel schéma de synchronisation classique cela se rapproche-t-il ? Pourquoi ?

▷ **Question 9:** Les porteurs d'eau perdent un temps précieux à contourner le bar et attendre leur tour au robinet. Proposez une solution à base de sémaphore(s) où l'un d'entre eux seulement est en charge de remplir les carafes pour les autres, qui courent ensuite en jeter le contenu dans les flammes quand elles sont pleines. De quel schéma de synchronisation classique cela se rapproche-t-il ? Pourquoi ?

▷ **Question 10: (question bonus)** Les porteurs d'eau, fatigués de courir en tout sens, décident d'organiser une chaîne en se passant les carafes pleines de proche en proche dans un sens, et les vides par le même chemin dans l'autre sens. Proposez une solution à base de sémaphore(s). De quel schéma de synchronisation cela se rapproche-t-il ? Pourquoi ?