

Aucun document n'est autorisé.

Exercice 1

1) Pour chacune des transitions suivantes entre les états des processus, indiquez si la transition est possible. Si c'est le cas, donnez un exemple.

- En exécution - prêt
- En exécution - bloqué
- Bloqué - en exécution
- En exécution - terminé

Exercice 2

1) Expliquez ce que fait ce programme C.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<unistd.h>

main(int argc, char *argv[])
{
    int attend=1,
    pos,
    debut;
    char * argument[20];

    if (argc==1){
        fprintf(stderr,"Usage : %s [-n] commande <arg1...>\n",argv[0]);
        exit(1);
    }

    if (strcmp(argv[1],"-n")==0){
        attend=0;
        debut=2;
    }

    else
        debut=1;

    pos=0;
    while(debut<argc){
        argument[pos] = (char * )malloc(strlen(argv[debut]));
        strcpy(argument[pos++],argv[debut++]);
    }
    argument[pos]=NULL;

    switch(fork()){
```

```

        case 0: execvp(argument[0],argument);
        case -1: printf("erreur fork\n");
                exit(-1);
        default: if (attend==1)
                wait();
                exit(0);
    }
}

```

Exercice 3 La RATP a écrit un programme qui simule le comportement des voyageurs aux portillons automatiques pour la sortie Maison des examens en gare de Sciences, et comptabilise ceux-ci afin d'étudier les flux de voyageurs en vue de rajouter des portillons à cette sortie.

Les règles de ce programme sont les suivantes :

- le nombre de voyageurs s'engageant vers la sortie Maison des Examens est géré par une variable `Nb-Voyageurs` qui est incrémentée dès qu'un voyageur s'engage dans l'un des escaliers des quais menant à cette sortie. Cette même variable est décrétementée dès que le voyageur a passé un portillon automatique et a quitté la station.
- Une autre variable `Nb-Voyageurs-Total` comptabilise le nombre total de voyageurs s'engageant vers la sortie Maison des Examens par jour.
- Un voyageur, pour sortir de la station, doit passer son ticket par un portillon et un seul et il doit être à un instant donné, le seul à utiliser ce portillon. Il y a 5 portillons permettant de sortir pour la sortie Maison des Examens.

On vous donne ici le pseudo-code d'un voyageur. Complétez ce code avec les opérations de synchronisation basées sur l'outil sémaphore que vous jugerez utiles. Vous recopierez le code sur votre copie.

Processus voyageur

Debut

Si (direction prise par le voyageur= sortie maison des examens)
alors

```
// compter un voyageur de plus pour cette sortie pour aujourd'hui
```

```
    Nb-Voyageurs-Total <-- Nb-Voyageurs-Total + 1;
```

```
// compter un voyageur de plus pour la sortie Maison des examens
```

```
    Nb-Voyageurs <-- Nb-Voyageurs + 1;
```

```
// attendre un portillon libre pour passer, puis passer
```

```
    Passer-portillon();
```

```
// compter un voyageur de moins pour la sortie Maison des examens
```

```
    Nb-Voyageurs <-- Nb-Voyageurs - 1;
```

sinon

Autre sortie de la station, on ne fait rien

Fsi

Exercice 4 Les politiques d'ordonnement sont sans réquisition (cas du DOS) ou avec réquisition (Cas d'Unix).

- 1) Expliquer sommairement la différence entre les deux.
- 2) Donner un exemple d'algorithme d'ordonnement pour chacune de ces politiques en expliquant les contraintes.