Nom : ……
Prénom : ……

Médian SR02
P2017
Durée de l'examen : 2h

- Répondez sur les feuilles de l'examen. Seules les réponses sur les feuilles de l'examen sont considérées.
- Les documents de cours et de TD (en version papier) sont autorisés

Partie I

Questions de compréhension de cours (5 pts)

1) Quel que soit le type du système d'exploitation (monoprogrammé ou multiprogrammé), chaque programme occupera le processeur à un certain moment. La multiprogrammation a été présentée comme un moyen pour améliorer l'efficacité de l'utilisation du processeur. Pourquoi le fait que le processeur ne fait pas les opérations d'E/S augmente les performances du système ? **(1 pt)**

……
……
……
……
……
……
……
……

2) Répondez par Vrai ou Faux aux questions suivantes et expliquez pourquoi :
   a) L'emplacement des registres est dans la RAM ? (**0.5 pt)**

……
……
……
……
……
……..

   b) Les opérations de lecture et d'écriture dans un pipe (tube) s'effectuent toujours de façon atomique ? (**0.5 pt)**

……
……

Nom : ……
Prénom : ……

……

……

……

……..

……..

3) Est-il possible d'exécuter un programme sans stocker ses données en mémoire ? Expliquer. (**0.5 pt)**

……

……

……

……

……

……

4) Y a-t-il une différence entre *un processus* et *un Job* ? Expliquer **(0.5 pt)**

……

……

……

……

……

……

……..

5) Dans quelques systèmes d'exploitation lorsqu'un processus crée un processus fils, l'exécution du processus créateur est suspendu jusqu'à la terminaison du processus fils, pourquoi ? **(1 pt)**

……

…..……..

…..……..

……

……

……

Nom : ……
Prénom : ……

6) Supposons qu'un processeur ne traite que les interruptions et qu'il a une capacité de lire ou d'écrire un mot mémoire de 4 octets en *10ns*, et que lors d'une interruption il faut recopier 32 registres dans la pile du processus. Quel est le nombre maximal d'interruptions que ce processeur peut traiter en *2ms* ? **(1 pt)**

……

……

……

……

……

……

……..

Exercice 1 (5 pts)

Question 1. En complétant le programme qui suit, écrire un programme qui a les fonctionnalités suivantes : Si vous l'exécutez sans paramètres sur la ligne de commande, il devra imprimer le contenu d'un segment de mémoire partagée. Sinon, si vous lui donnez un paramètre sur la ligne de commande, il devra stocker ce paramètre dans un segment de mémoire partagée.

```
#include<stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include<sys/shm.h>
……(1)……                    /* 1K de segment de mémoire
partagée */
int main(int argc, char *argv[]){
     ……(2)……
     ……(3)……
     ……(4)……
     int regilgen;
     if (argc> 2) {
          fprintf(stderr, "usage: arguments\n");
          exit(1);
     }
     /* Etablir la clé: */
     if (((cle = ……(5)……("fichier.c", 'L')) == -1) {
          ……(6)……
          ……(7)……
     }
     /* Connecter (et peut-être créer) le segment : */
```

Nom : ……
Prénom : ……

```
      if (……(8)……) {
            ……(9)……
            ……(10)……
      }
      /* Attacher au segment pour obtenir un pointeur sur celui-
ci:*/
      data=……..……(11)……
      if (……(12)……) {
            ……(13)……
            ……(14)……
      }
      /* Lire ou modifier le segment, en fonction de la ligne de
commande*/
      if (……(15)……)
            strncpy(data, argv[1], C);
      else
            printf("Le segment contient:\"……(16)….\"\n",.…(17)…);

      /*Commentaire caché*/

      if (……(18)……) {
            ……(19)……
            ……(20)……
      }
      return 0;
}
```

1 : ……

2 : ……

3 : ……

4 : ……

5 : ……

6 : ……

7 : ……

8 : ……

9 : ……

10 : ……

11 : ……

12 : ……

13 : ……

14 : ……

15 : ……

16 : ……

17 : ……

18 : ……

19 : ……

20 : ……

Nom : ……
Prénom : ……

Partie II

Exercice 2 (5 pts)

La suite de Syracuse est définie par :

$$u_{n+1} = \begin{cases} {}^{u_n}/_2 & si\ u_n\ est\ pair, \\ 3u_n + 1 & sinon \end{cases}$$

La conjecture de Syracuse affirme que, quel que soit le terme initial u_0 (entier strictement positif) de la suite, celle-ci finit par valoir 1 (puis boucler sur 4, 2, 1).
Etant donnée une valeur initiale u_0 de la suite, on appelle *temps de vol*, le plus petit indice k pour lequel $u_k = 1$ pour la première fois, et *altitude maximale* la valeur maximale prise par la suite durant ce temps de vol.

Exemple : pour $u_0 = 14$, on aura la suite des nombres suivants :

14, 7, 22, 11, 34, 17, **52**, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, **1**, 4, 2,1,4,2,1,4,2,1,4,2,1,4,2,1,4,2,1,4,2,1,4,2,1,…

Le temps de vol est donc de 17 ($u_{17} = 1$) et l'altitude maximale est de 52.

Question : écrire un programme qui exécute une boucle infinie dans laquelle à chaque itération, on fixe le terme initial u_0 de manière aléatoire, puis on calcule les termes de la suite de Syracuse jusqu'à ce qu'elle vaille 1. Le programme boucle indéfiniment sans rien afficher et a les fonctionnalités suivantes :

- A la réception du signal SIGTSTP (Ctrl-Z), le programme affiche :
  - la valeur initiale courante,
  - la valeur et l'indice du dernier terme calculé.
- A la réception du signal SIGINT (Ctrl-C), le programme affiche :
  - la plus grande valeur initiale testée,
  - le plus grand temps de vol atteint et la valeur initiale pour laquelle ce dernier est atteint,
  - la plus grande altitude maximale atteinte et la valeur initiale pour laquelle cette dernière est atteinte.

  Si le programme reçoit dans les trois secondes un autre signal SIGINT et si le signal SIGTSTP a été reçu au moins 10 fois, alors le programme se termine.

Remarque : il faut utiliser l'interface POSIX de gestion des signaux.

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

Nom : ……
Prénom : ……

/**********/
Variables globales **(0.25 pt)**
/**********/
……
……
……
……
……
……
……
……
……
……

/**********/***
Prototypes des fonctions **(0.25 pt)**
/**********/

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

Nom : ……
Prénom : ……

/**/
Programme principal **(1.5 pts)**
/**/
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

Nom : ……
Prénom : ……

/**********/

Définition des fonctions (**3 pts)**

/**********/

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Nom : ……
Prénom : ……

Partie III

Exercice 3 (2 pts)

Considérons un disque qui a 15 faces composées de 600 pistes et qui a une vitesse de rotation de 120 tours/seconde. Un secteur de ce disque a une taille de 1024 Mo. Dans une piste, on trouve 12 secteurs.

Question1 : Calculer la capacité totale du disque ainsi que le débit maximum du disque.

……..……

….……...……...……

….……...……...……

……...…...……

……...……

……...……...……

…...……...……

…...……...……

……

Question 2 : Calculer le nombre de pistes parcourues par la tête du disque avec chacun des algorithmes PCTR, FIFO, LOOK et SCAN pour la chaine de requêtes suivante :

5 - 17 - 2 - 200 - 90 - 115 - 85 - 3 - 210 - 550 - 600

On suppose qu'initialement la tête est à la piste 0.

……..……

….……...……...……

….……...……...……

……...…...……

……...……

……...……...……

…...……...……

…...……...……

……

……

……

……

……

Nom : ……
Prénom : ……

Exercice 4 (3 pts)

Soit une entrée/sortie de type imprimante installée sur un système à temps partagé. Un programme peut faire appel à cette imprimante à n'importe quel moment lors de son exécution pour imprimer un fichier de type FILE. Un programme peut choisir le nombre de copies à imprimer, en couleur ou en noir et blanc et enfin si l'impression se fera en recto-verso ou non. À la fin de chaque impression, un signal d'interruption est émis.

Question 1 : Quelles sont les structures nécessaires pour que ce système fonctionne correctement ?

…..……

…….……

……..……

……..……

……..……

……..……

……..……

……..……

Question 2 : Quels sont les programmes nécessaires dans cet exercice pour que le système fonctionne correctement ? Définir le comportement de chaque programme, l'enchainement et le lien entre chacun de ces programmes.

…..……

…….……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……

……..……