

	<p align="center">Examen Session Normale 11/01/2022</p>	
T. Garcia, J.A. Lorenzo	Système d'exploitation	
ING1 Informatique-Mathématiques appliquées	Année 2021–2022	

Modalités

- Durée : 2 heures.
- Vous devez rédiger votre copie à l'aide d'un stylo à encre exclusivement.
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Aucun document n'est autorisé.
- La calculatrice est autorisée.
- Aucune question ne peut être posée aux enseignants, posez des hypothèses en cas de doute.
- Aucune sortie n'est autorisée avant une durée incompressible d'une heure.
- Aucun déplacement n'est autorisé.

QCM à répondre sur votre copie et non pas sur le sujet : (6 points)

- a) La commande `cat /proc/<num>/maps`
- ☐ permet d'avoir la liste des "devices" nommés `num`.
 - ☐ permet de connaître la liste des processus.
 - ☐ permet de voir l'état de la mémoire pour le processus `num`.
 - ☐ Aucune des réponses n'est valide.
- b) Le fichier `/etc/apt/sources.list...`
- ☐ est mis à jour avec **`sudo apt-update`**.
 - ☐ définit les sources à partir desquelles `apt` va chercher les paquets.
 - ☐ est utilisé par l'utilitaire `dpkg` pour installer ou supprimer des paquets `*.deb`.
 - ☐ Aucune des réponses n'est valide.

- c) La commande **docker run -it ubuntu /bin/bash** exécutée deux fois d'affilée...
- ☐ Permet de créer deux conteneurs différents.
 - ☐ Permet de créer un conteneur dans la première exécution qui est redémarré pour la deuxième exécution.
 - ☐ Permet de créer deux conteneurs avec un volume partagé.
 - ☐ Aucune des réponses n'est valide.
- d) Un fichier...
- ☐ peut avoir plusieurs noms associés à plusieurs inodes.
 - ☐ est référencé par un numéro d'inode dans le système de fichiers.
 - ☐ possède une table de correspondance entre son nom et sa localisation sur disque.
 - ☐ Aucune des réponses n'est valide.
- e) Un dépôt PPA (*Personal Package Archives*) dans un système Ubuntu...
- ☐ est un dépôt de paquets logiciels dont le contenu est très ancien.
 - ☐ est ajouté à `/etc/apt/sources.list.d/` avec la commande `add-apt-repository`.
 - ☐ fait partie des dépôts officiels d'Ubuntu.
 - ☐ Aucune des réponses n'est valide.
- f) GPT...
- ☐ est un remplacement de l'UEFI.
 - ☐ est un remplacement de la BIOS.
 - ☐ est un standard de partitionnement.
 - ☐ Aucune des réponses n'est valide.

Questions de cours : (5 points)

- a) Expliquer les différences les plus importantes entre l'exécution d'un programme en mode kernel et en mode utilisateur.
- b) Expliquer la différence entre les gestionnaires de paquets **dpkg** et **apt**.
- c) Enumérer et expliquer chacune des régions allouées en mémoire lors du chargement d'un exécutable en UNIX.
- d) Qu'est-ce qu'un processus zombie ? Expliquer en détail son origine et comment il est géré par le système d'exploitation.

- e) Expliquer en détail pourquoi dans un environnement multiprogrammé il est nécessaire un mécanisme d'abstraction de la mémoire. Énumérer deux approches d'abstraction de la mémoire.

Exercice 1 : Programmation de processus (2 points)

Nous souhaitons faire un programme qui va créer quatre processus fils à partir d'un seul père. Après la création de tous les fils, le père doit attendre la terminaison de tous ses fils avant de finir lui-même. Nous sommes arrivés à écrire le code ci-dessous. Malheureusement, même si le code compile correctement, il contient deux erreurs qui empêchent son bon fonctionnement. Vous devrez :

- a) Trouver les deux erreurs dans le code.
b) Donner une sortie d'exemple du code corrigé.

```
1
2
3 #include <unistd.h>
4 #include <sys/types.h>
5 #include <stdio.h>
6 #include <sys/wait.h>
7 #include <stdlib.h>
8
9 int main() {
10     int retval, status, i;
11     int pidreturn = 0;
12     int tmp_p;
13     const int n = 4;
14     pid_t p[n];
15
16     printf("Pere: mon PID est %d\n",getpid());
17
18     for (i=0; i<n; i++)
19     {
20         p[i] = fork();
21         tmp_p = p[i];
22
23         if (p[i] != 0 )
24         {
25             printf("Fils: création du fils PID %d avec i=%d...\n",getpid(),i);
26             break;
27         }
28     }
29
30     if(tmp_p == 0){
31         do
32         {
33             printf("i: %d. Pere mis en attente. \n",i);
34             pidreturn = wait(&status);
35             printf("Pere: pidreturn %d terminé. Exit code est: %d\n", pidreturn, WEXITSTATUS(status));
36         }while(pidreturn != -1);
37     }else
38     {
39         sleep(10 - i);
40         printf("Fils PID %d. Valeur p[%d] = %d. \n",getpid(), i, p[i]);
41         exit(p[i]);
42     }
43 }
44
45
```

Exercice 2 : Le système de fichiers (2 points)

Considérons un système de fichiers similaire à celui d'UNIX. Supposant que :

- Le système de fichiers utilise des blocs de données de taille fixe 4K octets ;
- L'inode de chaque fichier contient 6 pointeurs directs sur des blocs de données, 1 pointeur indirect simple, 1 pointeur indirect double. Un pointeur indirect pointera sur une table à 96 entrées.
- Chaque pointeur (numéro de bloc) est représenté sur 2 octets (16 bits).

Répondre aux questions suivantes :

- a) Quelle est la plus grande taille de fichier que ce système de fichiers peut supporter ?
- b) Quelle est la taille du plus grand volume adressable ?
- c) Soit un fichier texte de 10000 caractères codés en ASCII (1 caractère = 1 octet). Combien de blocs de données sont nécessaires pour stocker ce fichier sur disque ? Schématiser le plan d'occupation des inodes.

Exercice 3 : Ordonnancement (3 points)

Considérons un système dans lequel les processus partagent un disque comme seul ressource autre que le processeur. **Cette ressource disque n'est accessible qu'en accès exclusif et non requérable**, c'est-à-dire qu'une commande disque lancée pour le compte d'un processus se termine normalement avant de pouvoir en lancer une autre. Un processus peut être en exécution, en attente d'entrées-sorties, en entrées-sorties ou en attente du processeur. Les demandes d'entrées-sorties sont gérées à l'ancienneté.

On considère 4 processus P1, P2, P3, P4 pour lesquels on sait que :

- Initialement, l'ordre de soumission des processus est P1, puis P2, puis P3, puis P4.
- Le processus élu à un instant t_0 est celui qui est le plus anciennement dans l'état prêt.

Les priorités des processus sont égales à 8 pour le processus P1, 6 pour le processus P2, 5 pour le processus P3 et 7 pour le processus P4. La plus grande valeur correspond à la priorité la plus forte.

Les 4 processus ont le comportement suivant :

P1 : Calcul pendant 40 ms, Lecture disque pendant 60 ms, Calcul pendant 10 ms, Lecture disque pendant 30 ms, Calcul pendant 10 ms

P2 : Calcul pendant 30 ms, Lecture disque pendant 30 ms, Calcul pendant 10 ms

P3 : Calcul pendant 20 ms, Lecture disque pendant 10 ms, Calcul pendant 10 ms

P4 : Calcul pendant 40 ms, Lecture disque pendant 10 ms, Calcul pendant 10 ms.

- a) On considère un système **monoprocesseur** dans lequel les processus partagent un disque comme seule ressource autre que le processeur et que l'ordonnancement sur le processeur se fait selon **une politique FIFO sans préemption**.

Remplissez le chronogramme d'exécution des 4 processus ci-dessus en tenant compte des états (prêt, élu (actif), en attente d'entrées-sorties et en entrées-sorties) en considérant que chaque case correspond à 10 ms.

- b) On considère un système **multiprocesseur composé de 2 cœurs** dans lequel les processus partagent un disque comme seule ressource autre que le processeur et que l'ordonnancement sur le processeur se fait selon **une politique FIFO avec préemption**.

Remplissez le chronogramme d'exécution des 4 processus ci-dessus en tenant compte des états (prêt, élu (actif) sur C1 ou sur C2, en attente d'entrées-sorties et en entrées-sorties) en considérant que chaque case correspond à 10 ms.

Exercice 4 : Mémoire virtuelle (2 points)

Un ordinateur possède un système de mémoire virtuelle avec pagination qui utilise des pages de 4 KB. Son espace de mémoire virtuelle est de 2^{32} bytes et la mémoire physique a une taille de 2^{18} bytes (256 KB). Répondre aux questions suivantes :

- a) Quel est le format de l'adresse virtuelle ? Indiquer chaque champ et le nombre de bits de chacun.
- b) Combien de cadres de page (frames) y-a-t-il dans la mémoire physique ?
- c) Quel est le nombre maximum d'entrées dans la table de pages ?