

Nom : BANET

Prénom : Loïc

N° de candidat :

Classe actuelle : ING1

## CONCOURS EPITA-IPSA 2015

Centre d'épreuve :

Épreuve : Système

Date : 13/01/2016

Note : \_\_\_\_\_ / \_\_\_\_\_

Appréciation : \_\_\_\_\_

1. Kernel : interface entre la machine et le logiciel.

Situé en round 0 (root, mode privilégié) le kernel est le cœur du système, il est présent dans tous les ordinateurs modernes.

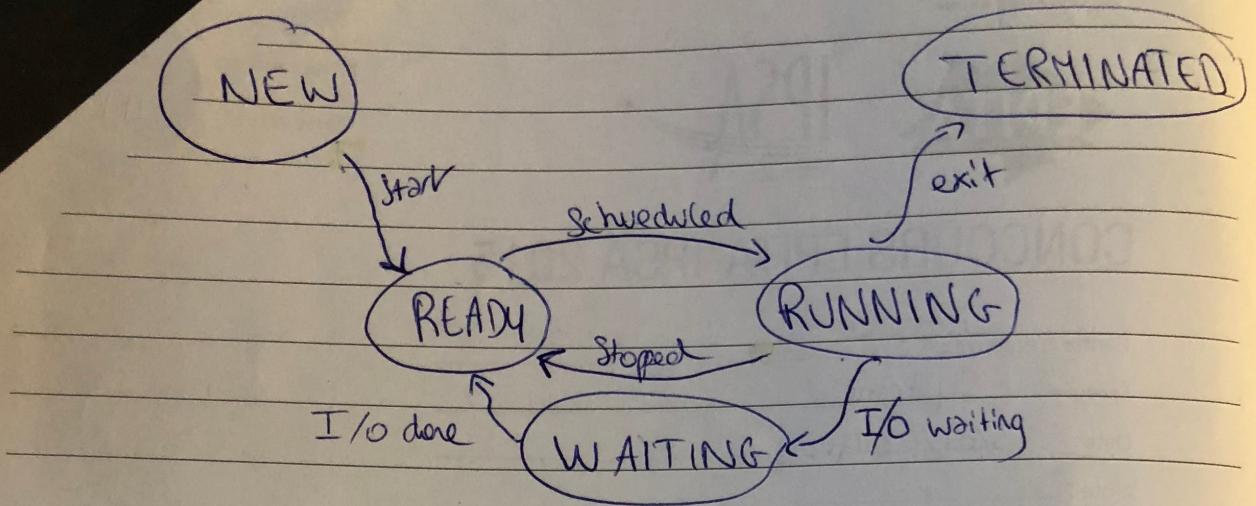
Services: Software, provided by the operating system, it ... is Input/outputs, critical error detection, ...

Librairies: Ensemble de process et méthodes utilisés pour réaliser des actions précises et pouvant être facilement réutilisées ex: libe pour utiliser toutes les fonctionnalités du langage c

Application: programme qui simplifie l'interaction entre l'utilisateur et la machine pour une action précise.

2- Les différents états des tâches d'un OS sont:

✓ NEW, READY, RUNNING, WAITING, TERMINATED



3- Round Robin: Il consiste à assigner le même temps d'exécution à chaque process. (Avec un temps de quantum fixe) Cela signifie que lorsque l'un des processus atteint son quantum, il est mis en pause dans la ready queue et le suivant est lancé. Cet algorithme peut être mis en pause. Les avantages sont une équité entre les processus par rapport au temps d'exécution de tous les processus, mais le round robin coûte cher en context switch. Car à chaque fois qu'on met en pause un processus, il y a un context switch (coûteux).

1. First Come First Serve: Cet algorithme consiste à assigner les processus dans leur ordre d'arrivée. Il est très simple et donc facile à implémenter. C'est une très bonne solution pour les systèmes batch mais est coûteux et très peu optimisé pour les autres systèmes.

Shortest Job First: Cet algorithme exécute les processus estimés les plus rapides en premier. Cela permet de réaliser beaucoup d'action très rapidement mais n'est pas toujours optimal.

pose un problème de starvation pour les programmes courteux. C'est à dire qu'il peut se passer un très long moment avant qu'ils soient exécutés ou ils peuvent également ne jamais être exécutés.

✓ 4. Les signaux sont des moyens simples de communication entre process dans les systèmes Unix. Ils agissent de manière asynchrone. Lorsqu'un signal est lancé, cela stoppe le fil d'exécution standard et donne priorité au signal.

SIGKILL : tue un processus, ne peut pas être catché

SIGSTOP : met le processus en pause, ne peut pas être catché

SIGALARM : reçoit une alarme liée à la fin d'un timer

SIGCONT : Continue le processus qui a été mis en pause par SIGINT

SIGINT : met en pause le processus qui peut être continué par SIGCONT

SIGABORT : tue un processus mais peut être catché

✓ 5- On peut exécuter une action whom quand on reçoit un SIGNAL avec SIGACTION.

les signaux qui ne peuvent pas avoir leur action par défaut réservé sont ceux qu'on ne peut pas catcher, c'est à dire SIGKILL et SIGSTOP.

✓ 6- Pour garantir une protection de la mémoire, la MMU, memory management unit va mapper un process dans une zone de mémoire virtuelle pour lui

faire croire qu'il est seul en mémoire (principe d'isolation)

✓ 7 - Afin de commencer un nouveau process, les systèmes devant être utilisés sont fork puis execve

8 - L'avantage d'utiliser les mêmes systèmes pour manipuler les fichiers et les appareils est que c'est plus simple, logique, compréhensible pour le programmeur, cependant il y a une perte de performance et de fonctionnalités quand on manipule les appareils avec les systèmes des fichiers, en effet certaines actions ne pourront pas être réalisées tandis que d'autres seront moins optimisées.

*Mostly true, but some are missing the point*

9- A = 0 (pid == 0)

B = 2603 (we are in the child so getpid() = 2603)

C = 2603 (Same logic)

D = 2600

10- A: value = 5 .

✓ The global variable value is shared between father and child process however, when modified in the child it is not modified in the father.

11- Components Shared across threads in a multithread process.

a- Register Values NO

b- Heap memory YES

c- Global variables YES

d- Stack memory NO

Nom : BANET  
Prénom : Loïc  
N° de candidat :  
Classe actuelle : ING 1

## CONCOURS EPITA-IPSA 2015

Centre d'épreuve :

Épreuve : Operating System

Date : 13/01/2016

Note : \_\_\_\_\_

Appréciation : \_\_\_\_\_

- 12 - I/O bound programs depends on an I/O return and aren't actually using much of the CPU resources, they usually are really slow.

CPU bound programs solely depends on the CPU and are really fast.

In order to ensure the best use of most of the CPU resources the scheduler must know which programs are CPU bound.

- 13 - Les sémaphores permettent de gérer les connexions comme une liste de file descripteurs. Lorsque trop de connexions simultanées sont détectées le système bloqe les ~~connexions~~ nouvelles connexions

**Nope**

- 14 - Un page faut se passer quand il y a un accès à

Une zone mémoire non mapée au process.  
Quand une page four arrive, le système envoie  
un SIGNAL SIGSEV qui bloque le process  
quoi?

8172  
 $\frac{u}{= 2048}$

15 - The maximum size of a file that can be stored  
in the file system is

$$\begin{aligned} & 4 \times 12 + \underbrace{8172 / 4 \times 4}_{\text{direct blocks}} + \underbrace{2048 \times 2048 \times 4}_{\text{single disk block}} + \underbrace{2048 \times 2048 \times 2048}_{\text{double disk block}} \times 4 \\ & = 48 + 8172 + 4193964 \times 4 + 4193964 \times 8172 \\ & = 8920 + 16775850 + 34355832800 \\ & = 34379616870 \text{ bytes soit environ } 34 \text{ TB} \end{aligned}$$

16 - ioctl est le syscall qui s'occupe des inputs claviers et devices. Ce syscall permet notamment de  $\rightarrow$  this is Read récupérer toutes les informations claviers en 1 seul syscall. Ce qui est pratique, peu coûteux et ne peut pas être réalisé aussi simplement avec des Read et Write.

Cependant ioctl ne fonctionne pas avec tous les devices et peut être un des points d'accès à des failles de sécurité, notamment dans les drivers des devices. Pour résoudre ce problème nous pourrions par exemple proposer une whitelist de drivers sécurisés que ioctl a le droit d'utiliser.