

Département math et Informatique

DUT Génie informatique

Module : Système d’exploitation

Semestre 2

Rapport mini projet : SIMULATION D’ALLOCATION DES CPU

**Réalisé par : Encadré par :**

**IMANE RACHID Professeur MEHDI ERRAJI**

**ESTE**

**2021/2022**

**REMERCIEMENT**

**On remercie Dieu .Tout puissant qui nous a donné l’opportunité de faire ce travail .C’est avec grand plaisir que nous adressons nos sincères remerciements à notre professeur Mehdi Erraji ,qui n’a ménagé aucun effort pour faire ce travail avec succès.**

***Sommaire :***

***1. Introduction ………………………………………………. Page 3***

* *Définition CPU………………………………………………****page 1***
* Processus …………………………………………… ***page 1-2***
* *Généralité processus …………………………………..* ***page 3***

***3. Allocation de CPU……………………………………….. Page 3***

* ***Le modèle FIFO………………………………….. page 5***
* ***Le modèle SJF et SRT……………………………… page 6-7***
* ***Le modèle Tourniquet……………………..page 8***

***4. Microprojet…………………………………………… page 9-13***

***5. Conclusion ………………………………………………... page 14***

1. **Introduction**

## Définition du CPU :

* Le **CPU** (Central Processing Unit), désigne la plupart du temps le [processeur](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445300-processeur-definition-fonctionnement-et-frequence/) d'un ordinateur. On peut le traduire en français par unité centrale de traitement (UCT) ou unité centrale de calculs.
* Le **CPU** a pour mission de réaliser les différents calculs inhérents au bon fonctionnement de l'ordinateur. Il constitue en ce sens un élément indispensable de la machine.

## Les processus :

* Un processus est une activité dynamique résultant l’exécution d’un programme avec ses données, par un processeur.
* C’est un programme au cours d’exécution.
* A quoi ça sert?
* À faire plusieurs activités ”en même temps”.
* Exemples :
* Faire travailler plusieurs utilisateurs sur la même machine. Chaque utilisateur a l’impression d’avoir la machine à lui tout seul.
* Compiler tout en lisant son mail

Problème: Un processeur ne peut exécuter qu’une seule instruction à la fois.

BUT: Partager un (ou plusieurs) processeur entre différents processus.

!!! Ne pas confondre processus et processeur.

Un processus est un dispositif électronique capable d’exécuter les instructions d’un processus. Il ne peut exécuter qu’une seule instruction à la fois.

3. Généralité Processus :

* Point de vue conceptuel: chaque processus possède son processeur virtuel.
* Réalité: le processeur bascule constamment d’un processus à l’autre.
* Ce basculement rapide est appelé multiprogrammation.
* Lorsque le processeur passe d’un processus à un autre, la vitesse de traitement d’un processus donné n’est pas uniforme et probablement non reproductible si le même processus s’exécute une nouvelle fois.

1. Allocation de CPU :

* Le modèle FIFO :

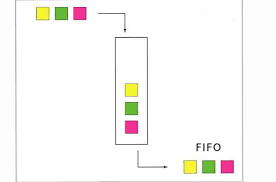


Figure 1 : diagramme Fifo

Principe :

* Chaque Processus possède un intervalle de temps, son quantum, pendant lequel il est autorisé à s’exécuter.
* Si le processus n’a pas encore terminé au bout de son quantum il est mis en fin de la liste des processus prêts et le processeur est réquisitionné et alloué à un autre processus.
* Rq: Application dans les systèmes à temps partagé où tous les processus utilisateurs possèdent la même priorité. Pour simplifier, nous supposerons que la tranche de temps allouée à chaque processus est constante.
* Le quantum est déterminé de tel sorte que le SE doit garantir un temps de réponse rapide (acceptable de 2 à 3s) pour les commandes chaque terminal,

!

* Le modèle SJF :

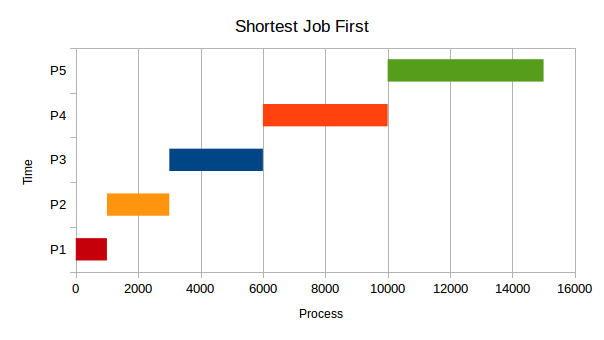


Figure 2 : diagramme SJF

* le modèle SJF (Shortest Job First) qui sert de façon prioritaire le processus qui demande le moins de temps au processeur. On suppose que cette information est connue avant l'exécution. Ce modèle est bien adapté à la description du traitement par lot où l'on doit préciser, lorsqu'on soumet une requête, le temps maximum de calcul. Le travail le moins demandeur est servi le premier.
* Le modèle SRT :
* Le modèle SRT (Shortest Remaining Time) le plus court temps restant (ou parfois Short remaining time first, le temps restant court en premier) est une Version préemptive du SJF. Elle donne la priorité aux processus dont le temps restant pour la fin d’exécution est court.

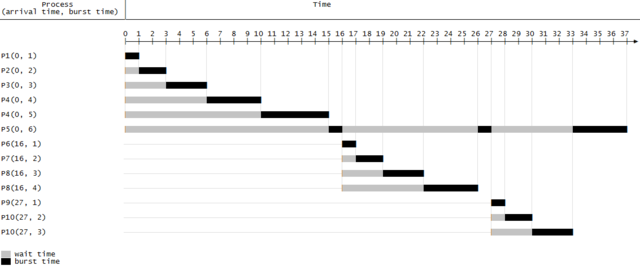


Figure 3: Diagramme SRT

* Tourniquet :
* Le modèle du tourniquet où il existe plusieurs files d'attente. Ce modèle représente assez bien une politique d'allocation du processeur pour le temps partagé, plus sophistiquée que le premier modèle évoqué.

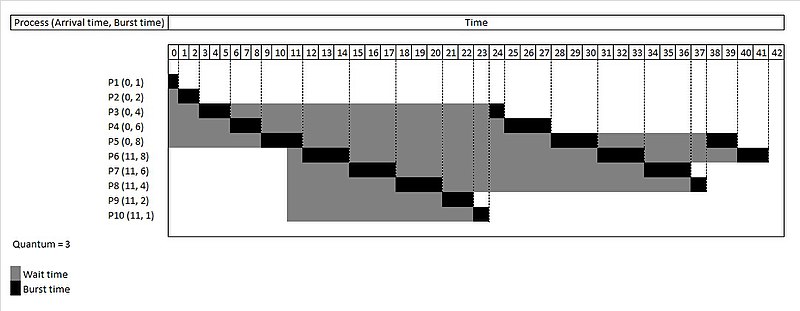


Figure 4: Diagramme Tourniquet

1. MICROPROJET :

1. Définir une description d’un processus (struct en langage C):

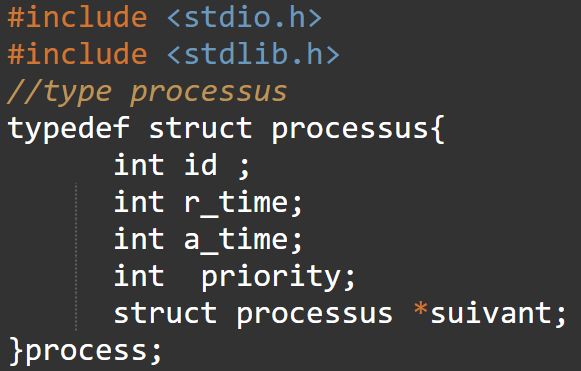
• identifiant

• temps restant

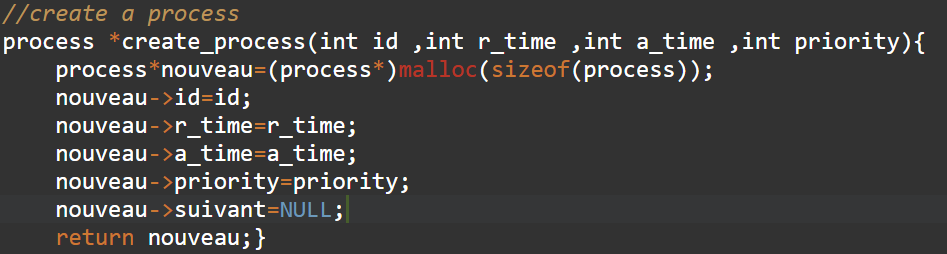
• date d’arrive;

• priorité

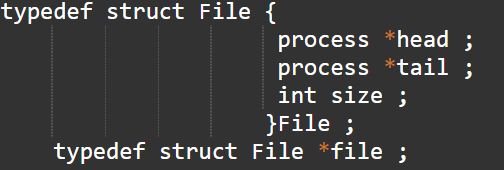
• l’adresse de processus suivant



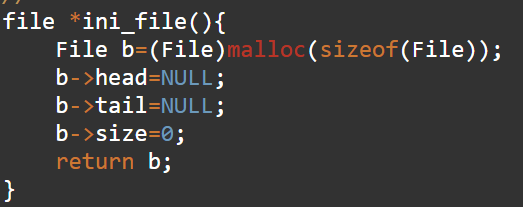
2. Ecrire une fonction en C qui nous permet de créer un processus : p r o c e s s ∗ c r e a t e\_ p r o c e s s ( i n t id , i n t r\_time , i n t a\_time , i n t p r i o r i t y ) ;



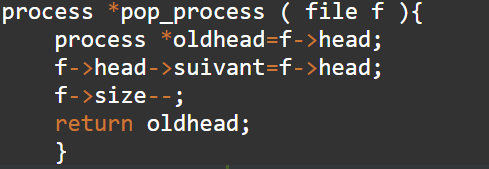
3. Propose une structure de file d’attente qui nous permet de stocker les processus en attente t y p e d ef s t r u c t F i l e { p r o c e s s ∗head ; p r o c e s s ∗ t a i l ; i n t s i z e ; } ; t y p e d ef s t r u c t F i l e ∗ f i l e ; Remarque : une variable de type file represent l’adresse d’une structure alloué dynamiquement de type File. Cela est fait pour garantir le passage par adresse dans les fonctions suivantes.



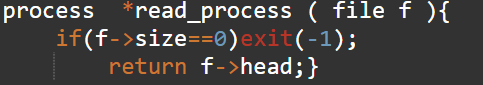
4. Propose une méthode pour initialiser une structure file. 1 f i l e i n i \_ f i l e ( ) ; ici, pour initialiser une file, il faut allouer la structure contenant la tête et le queue, les initialser par NULL et retourner l’adresse de cette dernière.



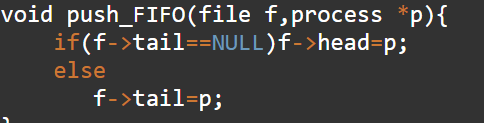
5. Propose une méthode pop qui permet de distribuer les processus en attente. p r o c e s s ∗ pop\_process ( f i l e f ) ;



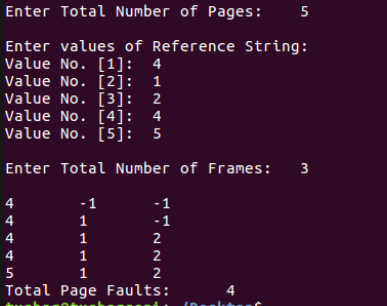
6. Définir une fonction qui nous permet d’examiner la tête de la file sans l’avoir consommé. p r o c e s s ∗ read\_p roce s s ( f i l e f ) ;



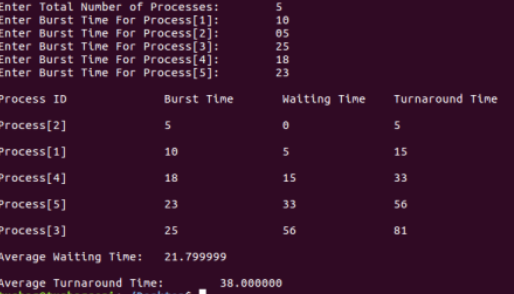
7. Selon chaque politique d’allocation, propose des fonctions push pour ordonnancer les processus. void push\_FIFO ( f i l e f , p r o c e s s ∗p ) ;



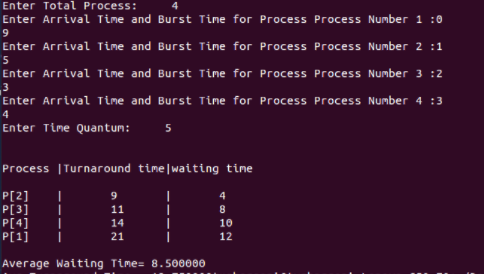
FIFO



SJF



\_ tou rnique t



1. Conclusion

Ce **microprojet**  a été très enrichissant pour moi, car il m'a permis de découvrir le domaine de système d’exploitation. Et j’ai appris comment trouver des solutions logiques.