

CY Paris Université

RAPPORT

pour le projet Système d'Exploitation
Licence d'Informatique troisième année

sur le sujet

Simulateur d'ordonnancement de processus

rédigé par

Da Cruz Mathis - Mian Farooq



Avril 2022

Table des matières

1	Introduction	3
1.1	Objectif du projet	3
1.2	Principe du projet	3
1.3	Objectif du rapport	3
2	Manuel d'utilisation	4

Table des figures

1	Écran affiché au lancement	4
2	Exemple de diagramme de Gantt FIFO	4
3	Exemple de diagramme de Gantt SJF	5
4	Exemple de diagramme de Gantt Round- Robin	5
5	Exemple de données fifo.csv	6
6	Exemple de données sjf.csv	6
7	Exemple de données rr.csv	6

Liste des tableaux

Remerciements

Les auteurs du projet tenaient à remercier sincèrement M.Laroque, qui en tant que professeur du module Systèmes d'Exploitation, s'est toujours montré à l'écoute et disponible pour notre groupe tout au long de ce semestre. En effet grâce à sa dévotion en tant que superviseur, il nous a fournis beaucoup d'aide durant les cours de TD. Nous remercions également toute l'équipe des professeurs de l'Université de Cergy pour nous avoir suivie et soutenue jusqu'ici.

1 Introduction

Dans cette section, nous allons présenter brièvement l'objectif du projet.

1.1 Objectif du projet

Le but du projet étant l'implantation d'un logiciel de simulation d'ordonnancement de processus dans un système d'exploitation. Pour ce faire, il faut utiliser au minimum les algorithmes FIFO, SJF, et round-robin. Nous devons donc créer une entrée de simulation (cycles E/S et CPU) ainsi qu'une sortie textuelle ou graphique.

1.2 Principe du projet

Un ordinateur possède plusieurs processus. Ainsi dans les systèmes d'exploitation, l'ordonnaceur est le composant du noyau de ce dernier choisant l'ordre d'exécution des processus. Par conséquent, l'algorithme d'ordonnancement permet l'identification du processus qui permettra la meilleure performance du système.

1.3 Objectif du rapport

Ce rapport peut se définir comme le guide de l'utilisateur. En effet, il précise comme l'algorithme doit être utilisé. Ce document pourra donc aider un utilisateur à se servir de notre projet.

2 Manuel d'utilisation

Tout d'abord, un logiciel d'environnement de développement doit être installé sur l'ordinateur comme par exemple : Visual Code. Ouvrez un terminal Linux et positionnez-vous a l'endroit de votre projet. Ensuite, l'utilisateur lance le programme via la commande "make" pour la compilation puis "./main" pour l'exécution. Une fois le programme exécuté, il sera demandé à ce dernier d'y entrer le nombre de processus qu'il souhaite ainsi que son temps d'arrivée (arrival time) et son temps d'éclatement (burst time) et ce pour chaque processus. Enfin, inséré la durée de quantum pour le round-robbin. Sur l'écran s'affichera donc :

```
Enter the no. of Process: 2

Enter the Details for Process 1:
    Arrival Time: 0
    Burst Time: 3

Enter the Details for Process 2:
    Arrival Time: 4
    Burst Time: 2

Enter the quantum:
2
```

FIGURE 1 – Écran affiché au lancement

Une fois cette saisie effectuée, l'utilisateur pourra alors voir le diagramme de Gantt pour chaque l'algorithme. Ainsi, que le détail pour chaque processus. À noter que l'affichage ne prend pas en compte les waiting time, ainsi comme on le voit ci-dessus le processus P2 arrivant à l'instant 4, débutera à l'instant 3 (l'endroit ou processus P1 prend fin) étant donné que le waiting time 3-4 n'est pas représenté.

```
<----- FIFO ----->

Gantt Chart:
|-----|-----|
|   P1   |   P2   |
|-----|-----|
0.0       3.0       6.0

Process Details:
Process 1:
Arrival Time: 0.00
Burst Time: 3.00
Turn Around Time: 3.00
Waiting Time: 0.00
Relative Delay: 1.00
Response Time: 0.00
Count : 3.00

Process 2:
Arrival Time: 4.00
Burst Time: 2.00
Turn Around Time: 1.00
Waiting Time: -1.00
Relative Delay: 0.50
Response Time: -1.00
Count : 6.00

Overall Details:
Throughput: 0.40
Average Turn Around Time: 2.00
Average Waiting Time: -0.50
Average Relative Delay: 0.75
Average Response Time: -0.50
```

FIGURE 2 – Exemple de diagramme de Gantt FIFO

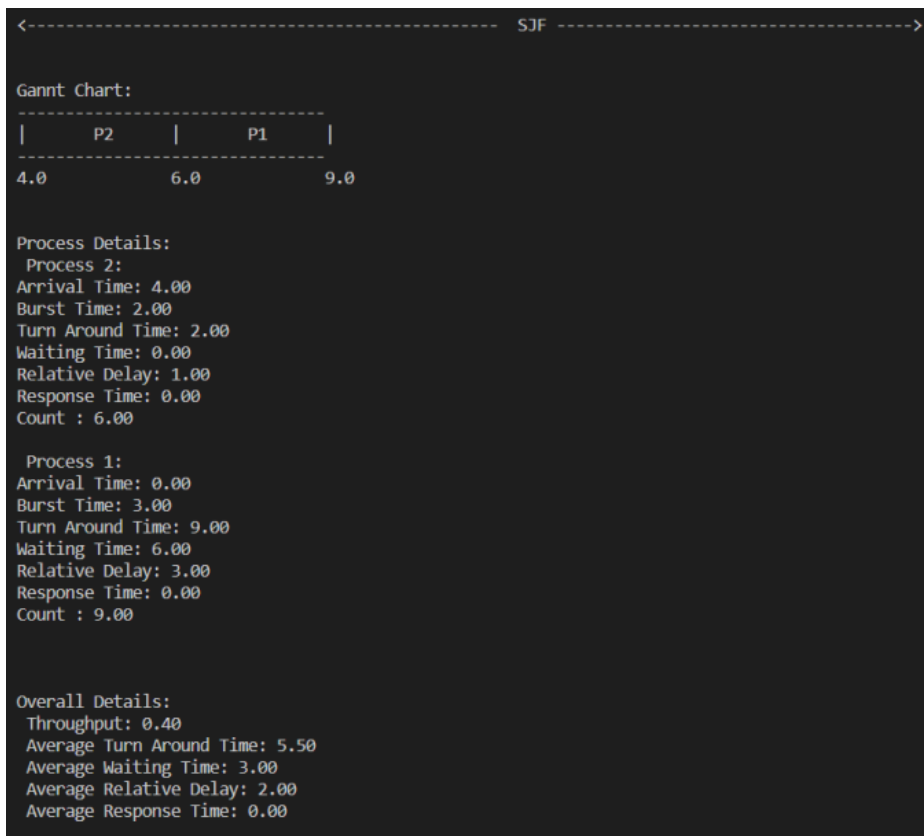


FIGURE 3 – Exemple de diagramme de Gantt SJF

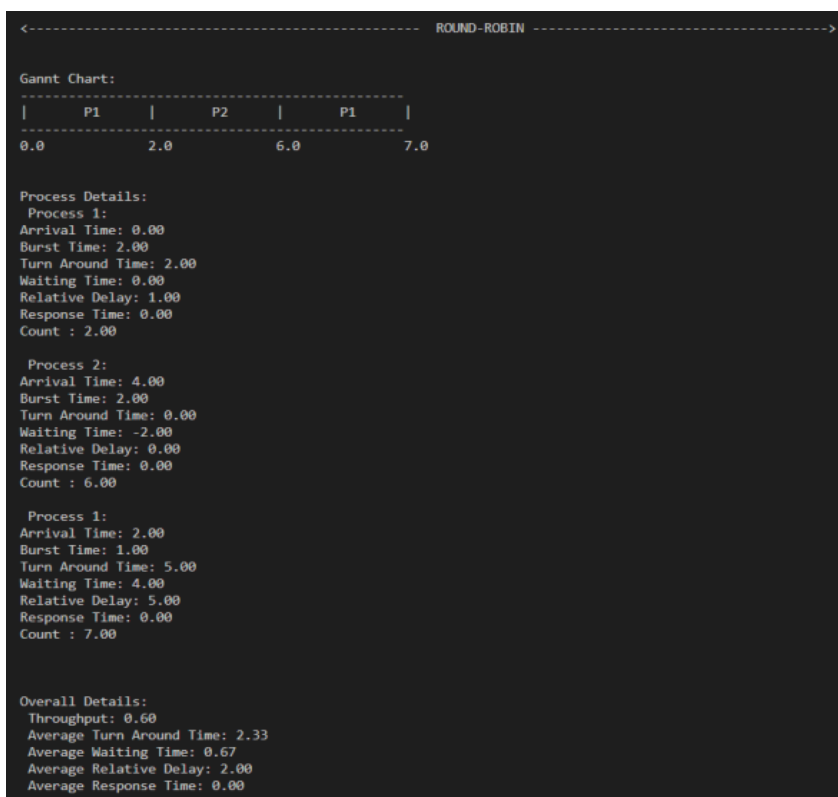


FIGURE 4 – Exemple de diagramme de Gantt Round- Robin

Chacun des détails ci-dessous seront affichés pour chaque processus. Ces données correspondent aux fichiers de configuration.

1. Temps d'arrivée
2. Temps d'éclatement
3. Délai d'exécution
4. Temps d'attente
5. Retard relatif
6. Temps de réponse

Une partie "overall details" est présente en fin de chaque ordonnecement. Elle permet de consulter les différents points décrits ci-dessous :

1. Throughput
2. Délai moyen d'exécution
3. Temps d'attente moyen
4. Temps relatif moyen
5. Temps de réponse moyen

L'utilisateur à la possibilité de récupérer l'ensemble des données de chaque processus pour chaque algorithme d'ordonnancement est stocké dans un fichier CSV dans le dossier nommé "excel". Ainsi il existe 3 fichiers CSV (fifo, sjf, round-robin).


```
excel >  fifo.csv
1 processus;arrival time;burst time;total arrival time;waiting time;relative delay;response time
2 1;0.000000;3.000000;6.000000;3.000000;2.000000;3.000000
3 2;4.000000;2.000000;9.000000;7.000000;4.500000;7.000000
```

FIGURE 5 – Exemple de données fifo.csv


```
excel >  sjf.csv
1 processus;arrival time;burst time;total arrival time;waiting time;relative delay;response time
2 2;4.000000;2.000000;2.000000;0.000000;1.000000;0.000000
3 1;0.000000;3.000000;9.000000;6.000000;3.000000;0.000000
```

FIGURE 6 – Exemple de données sjf.csv


```
excel >  rr.csv
1 processus;arrival time;burst time;total arrival time;waiting time;relative delay;response time
2 1;0.000000;2.000000;2.000000;0.000000;1.000000;0.000000
3 2;4.000000;2.000000;0.000000;-2.000000;0.000000;0.000000
4 1;2.000000;1.000000;5.000000;4.000000;5.000000;0.000000
```

FIGURE 7 – Exemple de données rr.csv

Enfin, il est toujours possible de nettoyer les compilations effectuées précédemment via les commandes "make clean"