

Projeto Prático

Sistemas Operacionais

Prof. Charles Ferreira

Leia atentamente as instruções

- Esta atividade prática pode ser realizada em grupo de até 4 alunos.
- O projeto deve ser realizado utilizando os conceitos apresentados em sala de aula;
- A entrega desta atividade deve ser realizada através do **Moodle**;
- Para enviar a atividade prática, o projeto deve ser compactado em um arquivo (zip, rar, tar, tgz, etc);
- Deve haver **instruções claras** de como executar o seu programa. Seja por linha de comando ou IDE.
- O prazo máximo para realizar o upload da atividade é o dia **06/11/2024** às **23:59** hs;
- Atividades enviadas **fora do prazo** estipulado **não serão aceitas**;
- Nos dias **07/11/2024** e **08/11/2024**, durante a aula, será realizada a avaliação através da apresentação e funcionamento do sistema;
- Linguagens permitidas: Java, Python, C ou C++;
 - Qualquer biblioteca externa utilizada (que não faça parte da distribuição comum da linguagem) deve ser mencionada e justificado a necessidade dessa biblioteca.
 - Se fizer em Python deverá ser entregue os arquivos .py do projeto. Nenhuma outra extensão de arquivo será aceita.
- Os itens que serão avaliados são:
 - (6,0) O funcionamento correto do sistema, atendimento as especificações e instruções aqui descritas;
 - (2,0) Aspectos inovadores que não constam na descrição desta atividade, enriquecendo o funcionamento do sistema;
 - (2,0) Respostas apresentadas durante a entrevista.
 - Trabalhos iguais serão **zerados**.
- **Observação importante:**
 - A apresentação tem caráter eliminatório, ou seja, **se não souber responder** as perguntas da apresentação seu trabalho inteiro poderá ser **zerado**.

Escalonamento Round-Robin

Esta atividade consiste na implementação de um programa que simule um escalonador Round-Robin preemptivo conforme os estudos realizados durante as aulas de Sistemas Operacionais.

O escalonador deve contemplar o funcionamento usual do algoritmo e também deve possuir a funcionalidade de haver operação de I/O que cada processo possa solicitar. Desta forma, o escalonador deve considerar o quantum de tempo que um processo pode utilizar a CPU, assim como quando o processo necessitar uma operação de I/O deve ser retirado de execução na CPU. Assim, todo processo que for retirado da CPU pelo escalonador, seja porque o quantum expirou ou por necessidade de uma operação de I/O, deve ser colocado no final da Fila de Pronto (fila de espera). Caso ocorra de um novo processo chegar no mesmo instante em que um processo que estava em execução foi retirado da CPU para a fila de espera, o novo processo terá prioridade em relação ao processo em execução para ir para a fila de espera. Caso haja operação de I/O, então o processo que solicitou I/O vai para a fila de prontos antes que o processo novo.

O simulador deve ter como entrada as informações de cada processo como PID, tempo de chegada, duração, e caso tenha operação de I/O, deve mostrar quando elas devem ser executadas (em relação ao seu tempo de execução). O tempo do quantum também deve ser descrito no início da simulação.

Após a leitura de dados, o simulador **deve apresentar em um arquivo separado** o resultado de execução dos processos em forma de um **diagrama de Gantt**, calculando o **tempo de espera** de cada processo e o **tempo de espera médio**.

Arquivo de entrada

A Figura 1 ilustra um exemplo de arquivo de entrada válido¹:

- Cada linha apresenta cada processo e suas características separadas por espaço em branco;
- A primeira palavra mostra o nome do processo (PID);
- A segunda palavra mostra o instante de chegada do processo;
- A terceira palavra mostra a duração do processo;
- Caso haja a quarta palavra, esta indicará se o processo realiza operações de I/O;
 - Caso o processo tenha operações de I/O, a palavra deve conter os instantes de I/O e os instantes estão separados por vírgula. Caso contrário a palavra pode ficar vazia.
- **Não serão fornecidos** outros arquivos de entrada para teste.

P1	10	9	2,4,6,8
P2	4	10	5
P3	0	5	2
P4	1	7	3,6
P5	17	2	

Figura 1: Exemplo de arquivo de entrada. Quantum = 4

¹Os projetos serão avaliados com arquivos diferentes

Arquivo de saída

A saída de dados **deve** ser realizada imprimindo o resultado em um arquivo de saída (**saida.txt**). O gráfico de Gantt **deve** ser gerado em um arquivo separado (**grafico.txt**). O exemplo a seguir ilustra os dados que devem ser apresentados para o arquivo de entrada da Figura 1.

```
*****
***** ESCALONADOR ROUND ROBIN *****
----- INICIANDO SIMULACAO -----
***** TEMPO 0 *****
FILA: Nao ha processos na fila
CPU: P3(5)
***** TEMPO 1 *****
#[evento] CHEGADA <P4>
FILA: P4(7)
CPU: P3(4)
***** TEMPO 2 *****
#[evento] OPERACAO I/O <P3>
FILA: P3(3)
CPU: P4(7)
***** TEMPO 3 *****
FILA: P3(3)
CPU: P4(6)
***** TEMPO 4 *****
#[evento] CHEGADA <P2>
FILA: P3(3) P2(10)
CPU: P4(5)
***** TEMPO 5 *****
#[evento] OPERACAO I/O <P4>
FILA: P2(10) P4(4)
CPU: P3(3)
***** TEMPO 6 *****
FILA: P2(10) P4(4)
CPU: P3(2)
***** TEMPO 7 *****
FILA: P2(10) P4(4)
CPU: P3(1)
***** TEMPO 8 *****
#[evento] ENCERRANDO <P3>
FILA: P4(4)
CPU: P2(10)
***** TEMPO 9 *****
FILA: P4(4)
CPU: P2(9)
***** TEMPO 10 *****
#[evento] CHEGADA <P1>
FILA: P4(4) P1(9)
CPU: P2(8)
***** TEMPO 11 *****
FILA: P4(4) P1(9)
CPU: P2(7)
***** TEMPO 12 *****
#[evento] FIM QUANTUM <P2>
FILA: P1(9) P2(6)
CPU: P4(4)
***** TEMPO 13 *****
FILA: P1(9) P2(6)
CPU: P4(3)
***** TEMPO 14 *****
```

```
FILA: P1(9) P2(6)
CPU: P4(2)
***** TEMPO 15 *****
#[evento] OPERACAO I/O <P4>
FILA: P2(6) P4(1)
CPU: P1(9)
***** TEMPO 16 *****
FILA: P2(6) P4(1)
CPU: P1(8)
***** TEMPO 17 *****
#[evento] OPERACAO I/O <P1>
#[evento] CHEGADA <P5>
FILA: P4(1) P1(7) P5(2)
CPU: P2(6)
***** TEMPO 18 *****
#[evento] OPERACAO I/O <P2>
FILA: P1(7) P5(2) P2(5)
CPU: P4(1)
***** TEMPO 19 *****
#[evento] ENCERRANDO <P4>
FILA: P5(2) P2(5)
CPU: P1(7)
***** TEMPO 20 *****
FILA: P5(2) P2(5)
CPU: P1(6)
***** TEMPO 21 *****
#[evento] OPERACAO I/O <P1>
FILA: P2(5) P1(5)
CPU: P5(2)
***** TEMPO 22 *****
FILA: P2(5) P1(5)
CPU: P5(1)
***** TEMPO 23 *****
#[evento] ENCERRANDO <P5>
FILA: P1(5)
CPU: P2(5)
***** TEMPO 24 *****
FILA: P1(5)
CPU: P2(4)
***** TEMPO 25 *****
FILA: P1(5)
CPU: P2(3)
***** TEMPO 26 *****
FILA: P1(5)
CPU: P2(2)
***** TEMPO 27 *****
#[evento] FIM QUANTUM <P2>
FILA: P2(1)
CPU: P1(5)
***** TEMPO 28 *****
FILA: P2(1)
CPU: P1(4)
***** TEMPO 29 *****
#[evento] OPERACAO I/O <P1>
FILA: P1(3)
```

```

CPU: P2(1)
***** TEMPO 30 *****
#[evento] ENCERRANDO <P2>
FILA: Nao ha processos na fila
CPU: P1(3)
***** TEMPO 31 *****
FILA: Nao ha processos na fila
CPU: P1(2)
***** TEMPO 32 *****
#[evento] OPERACAO I/O <P1>

```

```

FILA: Nao ha processos na fila
CPU: P1(1)
***** TEMPO 33 *****
#[evento] ENCERRANDO <P1>
FILA: Nao ha processos na fila
ACABARAM OS PROCESSOS!!!
-----
----- Encerrando simulacao -----
-----

```

Sugestões de extras

- Implementar **todos** os outros algoritmos de escalonamento vistos em aula (FIFO, SJF e Prioridade) com operação de I/O e fazendo uma comparação com o Round Robin em termos de tempo de espera e tempo de espera médio.
- Fazer uma apresentação de resultados de forma gráfica (pode ser no terminal) e em tempo real enquanto o algoritmo é executado. Sugestão: coloque um atraso (1 seg.) entre cada tempo. Não basta gerar uma imagem final da execução. Tem que ser gerado em tempo real, ou seja, enquanto executa vai mostrando o gráfico de gantt sendo feito.
- Acrescentar múltiplas CPUs. Neste caso, cada CPU tem sua fila de espera.
- Entre outros (consultar o professor).