

Atividade Prática de Sistemas Operacionais

Turma: Sistemas de Informação - 3º Período

Objetivo: Esta atividade tem por objetivo consolidar conceitos de processos e algoritmos de escalonamento.

Observações e regras de entrega:

- Esta atividade poderá ser realizada em **dupla**.
- Terá o valor de 2,5 pontos na composição da nota da Avaliação 2 (A2).
- A solução deverá ser enviada por meio da plataforma run.codes até as 23h59min do dia 11/08/2023.
- Pontos importantes a serem seguidos:
 - Conformidade com os requisitos aqui apresentados.
 - Uso dos conceitos de modularização.
 - Códigos bem estruturados e bem indentados.
- Caso seja detectada a ocorrência de plágio, será atribuída nota zero a **todos os envolvidos**.
- O não atendimento dos **requisitos especificados neste documento** acarretará em perda de pontos.

ATENÇÃO: Submissões que apresentarem erros de compilação ou falhas de segmentação (segmentation fault) no run.codes **NÃO** serão corrigidas.

Especificação

Como visto em aula, abaixo pode ser visto um diagrama com os estados genéricos de um processo em um Sistema Operacional.

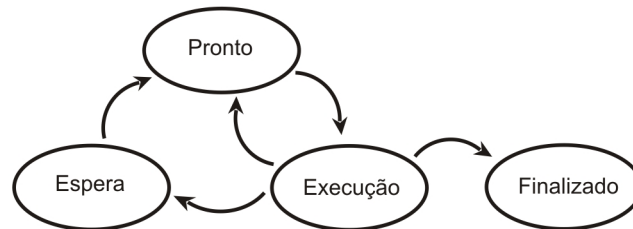


Figura 1: Esquema de estados de um processo

Tomando por base estes estados, você deverá criar um algoritmo capaz de simular um conjunto de processos que percorrem estes estados por meio de **Filas com Listas Encadeadas**. O seu algoritmo deverá funcionar da seguinte maneira:

- Trabalhar com a ideia de quantum de tempo (preempção) como visto nas aulas sobre algoritmos de escalonamento de processos. Este valor deve ser utilizado para controlar o tempo que cada processo fica executando antes de ser chaveado para outro processo ou ser finalizado.
- Deverão existir 4 filas de processos: Prontos, Espera, Execução e Finalizados, ou seja, uma fila para cada estado mostrado acima.
- Em cada fila deverá ser possível adicionar (enfileirar) e remover (desenfileirar) processos.
- A estrutura dos processos deverá conter 3 campos: Nome, Tempo de Execução (em décimos de segundo) e um ponteiro para o próximo processo.
- A estrutura das filas deverão possuir três campos: Nome, ponteiro para início e um ponteiro para o final.
- Você poderá fazer o uso da função `unsigned int sleep(unsigned int seconds);` fornecida pelo cabeçalho `unistd.h` para simular a passagem do quantum durante a execução de cada processo. Esta função poderá ser utilizada para testes locais do algoritmo.
- Para a versão a ser submetida no `run.codes`, você deverá utilizar a função `int usleep(useconds_t usec);` também disponível no cabeçalho `unistd.h`.
- Todos os dados referentes ao quantum e respectivos processos deverão ser lidos de um arquivo.
- O Programa deverá ler o caminho do arquivo no início da execução através de um simples `scanf`.
- Após a leitura, os dados dos processos deverão ser inseridos na fila de prontos.

- O arquivo de entrada estará sempre no seguinte formato:

```
<valor_quantum>
<P1>\t<temp_exec1>
<P2>\t<temp_exec2>
<P3>\t<temp_exec3>
:
<Pn>\t<temp_execn>
```

Desta forma, um exemplo de arquivo válido seria o seguinte:

```
3
P1  10
P2   5
P3   3
P4   6
```

- Logo após a leitura e inserção na fila de prontos, os processos deverão ser inseridos na fila de execução e a simulação será iniciada com o primeiro processo executando:
 - Se a fatia de tempo (quantum) expirou e ainda restar algum tempo de execução no processo, este deverá ser adicionado a fila de espera, caso contrário deverá ser adicionado a fila de finalizados. Este comportamento se dará enquanto existirem processos na fila de execução.
 - Quando a fila de execução estiver vazia e ainda existirem processos na fila de espera, estes deverão ser movidos para a fila de prontos, e posteriormente para a fila de execução.
 - Deste modo a execução se inicia novamente até que todos os processos sejam adicionados a fila de finalizados, quando então a simulação será finalizada.
- O estado atual das filas de processos deverá ser mostrado todas as vezes em que ocorrer alguma mudança nos processos/filas:
 - Transição de estados de processo: execução → espera e execução → finalizados.
 - Movimentação nas filas: espera → prontos e prontos → execução.
- Seu programa deverá liberar toda e qualquer memória previamente alocada dinamicamente.

Exemplo de execução do algoritmo:

Suponha que o arquivo citado acima tenha sido lido, então a saída de seu algoritmo deverá ser a seguinte:

```
./exercicio.out
dados_processos.dat
```

Estado atual das filas de processos:

Pronto: $P1 = 10 \leftarrow P2 = 5 \leftarrow P3 = 3 \leftarrow P4 = 6$

Espera: vazia

Execucao: vazia

Finalizados: vazia

Quantum: 3

----- > Iniciando execucao

Processos foram adicionados a fila de execucao

Estado atual das filas de processos:

Pronto: vazia

Espera: vazia

Execucao: $P1 = 10 \leftarrow P2 = 5 \leftarrow P3 = 3 \leftarrow P4 = 6$

Finalizados: vazia

P1 esta executando...

Quantum expirou, P1 sofreu preempcao

P1 foi adicionado a fila de espera

Estado atual das filas de processos:

Pronto: vazia

Espera: $P1 = 7$

Execucao: $P2 = 5 \leftarrow P3 = 3 \leftarrow P4 = 6$

Finalizados: vazia

P2 esta executando...

Quantum expirou, P2 sofreu preempcao

P2 foi adicionado a fila de espera

Estado atual das filas de processos:

Pronto: vazia

Espera: $P1 = 7 \leftarrow P2 = 2$

Execucao: $P3 = 3 \leftarrow P4 = 6$

Finalizados: vazia

P3 esta executando...

P3 terminou a execucao, P3 foi adicionado a fila de finalizados

Estado atual das filas de processos:

Pronto: vazia

Espera: $P1 = 7 \leftarrow P2 = 2$

Execucao: $P4 = 6$

Finalizados: $P3 = 0$

P4 esta executando...

Quantum expirou, P4 sofreu preempcao

P4 foi adicionado a fila de espera

A fila de execucao esta vazia, movendo processos para a fila de prontos

Estado atual das filas de processos:

Pronto: $P1 = 7 \leftarrow P2 = 2 \leftarrow P4 = 3$

Espera: vazia

Execucao: vazia

Finalizados: $P3 = 0$

Processos foram adicionados a fila de execucao

Estado atual das filas de processos:

Pronto: vazia

Espera: vazia

Execucao: $P1 = 7 \leftarrow P2 = 2 \leftarrow P4 = 3$

Finalizados: $P3 = 0$

P1 esta executando...

Quantum expirou, P1 sofreu preempcao

P1 foi adicionado a fila de espera

Estado atual das filas de processos:

Pronto: vazia

Espera: $P1 = 4$

Execucao: $P2 = 2 \leftarrow P4 = 3$

Finalizados: $P3 = 0$

P2 esta executando...

P2 terminou a execucao, P2 foi adicionado a fila de finalizados

Estado atual das filas de processos:

Pronto: vazia

Espera: $P1 = 4$

Execucao: $P4 = 3$

Finalizados: $P3 = 0 \leftarrow P2 = 0$

P4 esta executando...

P4 terminou a execucao, P4 foi adicionado a fila de finalizados

A fila de execucao esta vazia, movendo processos para a fila de prontos

Estado atual das filas de processos:

Pronto: $P1 = 4$

Espera: vazia

Execucao: vazia

Finalizados: $P3 = 0 \leftarrow P2 = 0 \leftarrow P4 = 0$

Processos foram adicionados a fila de execucao

Estado atual das filas de processos:

Pronto: vazia

Espera: vazia

Execucao: $P1 = 4$

Finalizados: $P3 = 0 \leftarrow P2 = 0 \leftarrow P4 = 0$

P1 esta executando...

Quantum expirou, P1 sofreu preempcao

P1 foi adicionado a fila de espera

A fila de execucao esta vazia, movendo processos para a fila de prontos

Estado atual das filas de processos:

Pronto: $P1 = 1$

Espera: vazia

Execucao: vazia

Finalizados: $P3 = 0 \leftarrow P2 = 0 \leftarrow P4 = 0$

Processos foram adicionados a fila de execucao

Estado atual das filas de processos:

Pronto: vazia

Espera: vazia

Execucao: $P1 = 1$

Finalizados: $P3 = 0 \leftarrow P2 = 0 \leftarrow P4 = 0$

P1 esta executando...

P1 terminou a execucao, P1 foi adicionado a fila de finalizados

Estado atual das filas de processos:

Pronto: vazia

Espera: vazia

Execucao: vazia

Finalizados: $P3 = 0 \leftarrow P2 = 0 \leftarrow P4 = 0 \leftarrow P1 = 0$

Nao ha mais processos a serem executados

----- > Simulacao finalizada

Bom trabalho!