

Trabalho 4: Escalonador de processos

Trabalho individual

Considere uma fila de processos a serem escalonados por um sistema operacional. Cada processo possui as seguintes informações: id, tempo de chegada, tempo de execução e tipo (de *sistema* ou de *usuário*). O escalonamento funciona da seguinte forma:

Processos vão sendo colocados em espera na ordem em que forem chegando ao processador (de acordo com o tempo de chegada). Os processos são informados pelo usuário. Após o encerramento da entrada de dados, processos são removidos da espera e repassados para o processador. O processo fica no processador por um tempo fixo denominado de TP (informado pelo usuário e válido para todos os processos). Se o tempo de execução for maior que TP, o processo processa por um período de tempo TP e volta para o final da fila de processos em espera. Caso contrário, ele executa o tempo especificado de execução e libera o processador. Use uma fila para armazenar os processos e simular tal escalonador.

Os processos podem ser de dois tipos: de sistema ou de usuário. Processos de sistema sempre têm prioridade para executar se comparados com os processos de usuário. Sempre que o processador estiver livre, a fila de processos deve ser percorrida de forma a verificar se não existem processos de *sistema* esperando. Nesta busca, os processos de usuário devem ser movidos temporariamente para uma estrutura auxiliar (pilha), até que seja encontrado algum processo de SO (sistema) ou até a fila de espera ficar vazia. Caso exista um processo de SO, este deve ser enviado ao processador e os outros processos de usuário na estrutura auxiliar (pilha) devem ser recolocados novamente na ordem anterior dos processos. Simule tal escalonador usando pilhas e filas.

Dicas:

- Use um contador para simular a passagem de tempo.
- Se preciso for, use mais de uma estrutura auxiliar.
- Podem usar filas duplas, se necessário.

O programa deve receber como entrada:

- um conjunto de processos a processar (para cada processo, informar id, tempo de chegada, tempo de execução e tipo (de *sistema* ou de *usuário*);
- TP: tempo fixo que o processo pode ficar no processador;

O programa deve gerar na saída:

- a ordem de execução dos processos (mostrar o id), juntamente com o tempo em que ficaram executando
 - * cada vez que os processos forem empilhados (na busca por processos de sistema), deve-se mostrar o conteúdo da pilha gerada). Ver exemplo a seguir:

1) Exemplo:

Entrada: formato <id, tempo de chegada, tempo de execução e tipo>

P1, 1, 20, usuário

P2, 2, 10, usuário

P3, 3, 30, usuário

P4, 4, 15, usuário

TP: 20

Procurando por processo de sistema:

Conteúdo da pilha (a partir do topo): P4 (topo), P3, P2, P1

Não achou processo de sistema, recoloca eles na fila, mantendo a ordem correta de chegada.

Conteúdo da fila: P1, P2, P3, P4

Processa o primeiro da fila: P1 (20)

Fila passa a ter: P2, P3, P4

Procurando por processo de sistema:

Conteúdo da pilha: P4 (topo), P3, P2

Não achou processo de sistema, recoloca eles na fila, mantendo a ordem correta de chegada.
Conteúdo da fila: P2, P3, P4

Processa o primeiro da fila: P2 (10)

Fila passa a ter: P3, P4

Procurando por processo de sistema:

Conteúdo da pilha: P4 (topo), P3

Não achou processo de sistema, recoloca eles na fila, mantendo a ordem correta de chegada.

Conteúdo da fila: P3, P4

Processa o primeiro da fila: P3 (20) – ainda fica faltando processar mais 10, insere P3 ao final da fila

Fila passa a ter: P4, P3

Procurando por processo de sistema:

Conteúdo da pilha: P3 (topo), P4

Não achou processo de sistema, recoloca eles na fila, mantendo a ordem correta de chegada.

Conteúdo da fila: P4, P3

Processa o primeiro da fila: P4 (15)

Fila passa a ter: P3

Procurando por processo de sistema:

Conteúdo da pilha: P3

Não achou processo de sistema, recoloca eles na fila, mantendo a ordem correta de chegada.

Conteúdo da fila: P3

Processa o primeiro da fila: P3 (10)

Fila passa a ter: -

Ordem de execução: P1 (20), P2(10), P3(20), P4(15), P3(10)

2) Outro exemplo:

Entrada: formato <id, tempo de chegada, tempo de execução e tipo>

P1, 1, 20, usuário

P2, 2, 10, sistema

P3, 3, 10, usuário

TP: 20

Procurando por processo de sistema:

Conteúdo da pilha: P1

Achou P2, que é processo de sistema. Executa **P2 (10)**, e recoloca os processos da pilha na fila, mantendo a ordem correta de chegada. Conteúdo da fila: P1, P3

Fila fica com P1 e P3, ambos de usuário. Então processa P1 e P3, nesta ordem.

Ordem de execução: P2(10), P1(20), P3(10)

Entrega: por e-mail para deise@inf.ufsm.br os arquivos pertencentes ao projeto em questão. Não inclua os arquivos .exe.

Data da entrega: 02 de junho

Observações para este trabalho:

- Considere as *instruções gerais* disponíveis no site da disciplina.
- As *structs* devem sempre ser alocadas dinamicamente.
- O código deve ser organizado em diferentes arquivos e funções.
- Cópias da internet e/ou colegas anulam a nota do trabalho.
- Trabalhos com erros de compilação não serão corrigidos.