

Estrutura de Dados - ARA 7125

Segundo Exercício-Programa - Entrega: 1 de Julho de 2015

Escalonador de Processos Simplificado

Este Exercício-Programa consiste na implementação de um sistema de escalonamento de processos simplificado. **A estrutura de dados fila de prioridade implementada em um max-heap deve ser usada. O max-heap também deve ser implementado em um vetor.**

Um processo pode ser visto como um conjunto de instruções que deve ser executado pelo processador de uma máquina. A cada instante de tempo, um sistema de escalonamento de processos deve escolher um processo para ser executado por um processador. Cada processo tem o seu identificador único (formado pelo prefixo “id” seguido por um número inteiro entre 1 e 5000), o tempo necessário para executar todas as suas instruções no processador (um número inteiro entre 1 e 100 (unidade em milissegundos)), e a prioridade do processo (um número inteiro entre 1 e 1000). A cada instante de tempo, ou o processador fica ocioso (não existe processo esperando na fila para ser executado), ou um novo processo é escolhido espontaneamente para ser executado (termina a execução de um processo), ou um processo é escolhido abruptamente para ser executado (passaram 50 ms de execução de um processo). Segundo essa política, note que um processo que estava em execução pode ser forçado a parar e, se este processo ainda possuir a maior prioridade, ele deve ser escolhido novamente para ser executado. Resumindo, a cada instante de tempo, o sistema de escalonamento deve:

1. atualizar o tempo necessário de execução do processo que estava em execução;
2. incluir na fila de prioridade o processo que estava em execução (se for necessário);
3. incluir na fila de prioridade os novos processos (se existir); e
4. remover da fila de prioridade um processo de maior prioridade para ser executado (se existir).

A entrada do seu programa será composta por um número inteiro n ($n > 0$) seguido por n linhas com as seguintes informações (para cada linha): o número de processos que desejam ser executados, o identificador de cada processo, a prioridade de cada processo, e o tempo necessário de processamento para cada processo.

Exemplo de entrada:

```
4
3 id1 id2 id3      44 15 60      55 29 15
4 id4 id5 id6 id7  61 32 46 11   11 13 54 64
0
2 id8 id9          22 29          35 89
```

A interpretação da segunda linha da entrada é: neste instante de tempo, chegaram 3 processos no sistema identificados por id1, id2, id3, com prioridades 44, 15 e 60 (respectivamente) e tempos necessários de processamento 55, 29 e 15 milissegundos (respectivamente). O processamento da entrada termina depois que a última linha é processada.

A cada instante de tempo, imprima na saída padrão (tela do computador) o identificador do processo escolhido para ser executado (se existir algum). Depois do processamento de toda a entrada, imprima na saída padrão as informações dos processos que estão no max-heap final, na ordem crescente dos índices do vetor onde foi implementado o max-heap.

Observações importantes:

1. Podem formar grupos com até 3 pessoas.
2. O seu programa deve ser feito em linguagem C. O compilador deve ser o gcc.
3. Você pode usar os algoritmos vistos em sala de aula.
4. Exercícios-Programas atrasados **não** serão aceitos.
5. Programas com erros de sintaxe (ou seja, existem erros durante a compilação do programa), receberão nota ZERO. Programas com *warning* na compilação terão diminuição na nota.
6. Utilize apenas os recursos que foram vistos em aula.
7. É importante que seu programa esteja escrito (digitado) de maneira a destacar a estrutura do programa (boa formatação).
8. O seu programa deve começar com um cabeçalho (linhas em comentários) contendo pelo menos o seu nome, sua matrícula e seu curso.
9. Coloque comentários em pontos convenientes do programa.
10. A entrega do Exercício-Programa deverá ser feita no MOODLE.
11. O Exercício-Programa é individual por grupo. Não copie o programa de outro grupo, não empreste o seu programa para outro grupo, e tome cuidado para que não copiem seu programa sem a sua permissão. Todos os programas envolvidos em cópia terão nota ZERO.