Universidade Federal do Ceará
Disciplina: Sistemas Operacionais
Prof. Ms. Rafael Ivo

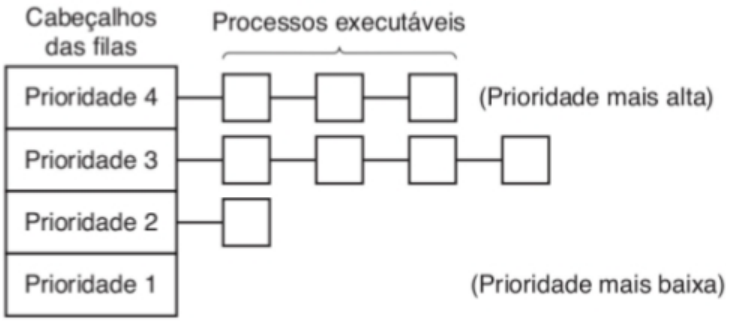
Trabalho 1: Escalonamento por Prioridades

Regras gerais:

- O trabalho deve ser em grupos de até 3 pessoas. Os grupos devem ser formados até dia 05/09 (quarta-feira) e informados ao professor após o término da aula. Não será permitida a inserção de novo(s) integrante(s) em um grupo formado após esta data.
- O envio do trabalho será exclusivamente no SIGAA, no prazo estipulado.
- O código deve ser muito bem comentado, explicando as ideias utilizadas. Deve ser possível entender todo o funcionamento do código apenas lendo os comentários.
- O início do código deve conter o nome dos integrantes do trabalho. Se o código possuir mais de um arquivo, a identificação deve estar em cada um dos arquivos.

Descrição:

Um dos algoritmos de escalonamento de processos mais utilizado pelos sistemas operacionais interativos é o algoritmo de prioridades. A ideia básica é simples: a cada processo é atribuída uma prioridade e ao processo pronto com maior prioridade é permitido executar. Os processos são divididos em filas de prioridade. Caso dois processos possuam a mesma prioridade, seu funcionamento é idêntico ao algoritmo de Round-Robin.



Seu programa irá simular o algoritmo de escalonamento usando threads. No primeiro instante, ele deve pedir as seguintes informações do usuário:

- Quantas threads você deseja criar?
- Qual a prioridade de cada thread?
- Quantas instruções cada thread deseja executar antes de pedir uma E/S (sair para estado de bloqueado)?
- Quanto tempo cada thread passa esperando E/S (voltar para o estado de pronto)?
- Qual o valor do quantum (tempo máximo que uma thread usa o processador)?
- Depois de quanto tempo a simulação deve acabar?

O papel de cada thread é simplesmente possuir uma variável local incrementando e escrevendo: “Sou a thread X e estou executando a instrução N”. Deve-se manter uma thread central, o escalonador, para interromper uma thread e liberar outra para executar. Não deve ser permitido que duas threads criadas executem ao mesmo tempo. Este escalonador deve fazer uso de semáforos para controlar as threads do usuário e possuir internamente variáveis para controlar de quem é a vez de executar.

Avaliação:

Será avaliado se a técnica foi implementada usando os conceitos do algoritmo, ou seja, sem “gambiarras” ou que funcione apenas em casos específicos. O código deve estar claro e entendível. Ao final do semestre, durante a apresentação dos trabalhos, este código será explicado pelos integrantes da equipe. Todos do grupo devem explicar uma parte específica e terão suas notas de apresentação individualmente. A nota dada pela implementação é idêntica a todos. A nota do trabalho será a média destas duas notas. Trabalhos copiados receberão nota zero, indepentende se comprove ser o trabalho original. A nota do trabalho vale 25% da nota da Avaliação Parcial 1.