

Simulador de Escalonamento

Considere um sistema com uma CPU e um dispositivo de I/O. O sistema possui o parâmetro Grau de Multiprogramação (ML), o qual determina o nº máximo de processos existentes. A CPU possui uma *fila de prontos* (**ready queue**) e o dispositivo de I/O possui uma *fila de dispositivos* (**I/O queue**). A carga inicial do sistema e, consequentemente, a manutenção do grau de multiprogramação é dada por uma fila denominada *fila de chegada* (**arrive queue**). De tempos em tempos, o programa gera processos que irão executar no sistema (simulando o escalonador de longo prazo) e os coloca na fila de chegada (ordem FIFO). Se o sistema estiver com a carga completa (nº de processos no sistema = ML), o processo ficará esperando, caso contrário, entrará no sistema.

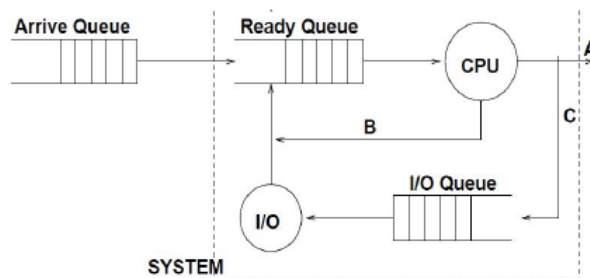


Figura 1: Fluxo de execução do Simulador

Uma vez no sistema, o processo irá tentar obter a CPU. Se a CPU estiver ocupada por outro processo, o processo solicitante é colocado na fila de prontos (ordem FIFO). Caso contrário, o processo obtém a CPU e executa até que uma das 3 situações ocorra:

- Processo concluído (fluxo A na Figura 1) - O processo executou na CPU pelo período necessário para sua conclusão e pode ser removido do sistema. O primeiro processo da fila de chegada deve ser inserido no sistema. Se não houver processo na fila de chegada, deve-se anteriormente disparar geração de processos.
- Término da fatia de tempo (fluxo B na Figura 1) - O processo é interrompido em função do fim de sua fatia de tempo. O processo é retirado da CPU, colocado ao final da fila de prontos e sua fatia de tempo é resetada.
- Solicitação de I/O (fluxo C na Figura 1) - O processo solicitou I/O. O processo é retirado na CPU e alocado ao dispositivo de I/O. Se o dispositivo estiver ocupado, o processo aguarda na fila de dispositivos, a qual está organizada em ordem de prioridade e implementada por meio de uma lista duplamente encadeada. Ao término da operação de I/O, o processo retorna para a fila de prontos.

Nas 3 situações, a CPU é liberada pelo processo atual e a fila de prontos é examinada a fim de escolher o próximo processo a executar. O algoritmo de escalonamento é o **Round Robin**, no qual uma fatia de tempo determina a interrupção do processo e o primeiro processo da fila de prontos será o próximo a ser executado (execução circular).

Quando inicializado, o simulador deve ler os seguintes valores inteiros de um arquivo de entrada:

- Tempo total da simulação
- Grau de Multiprogramação
- Fatia de tempo para escalonamento RR

O formato do arquivo de entrada (.txt) consiste nos três valores acima, um por linha.

Os seguintes parâmetros devem ser gerados aleatoriamente pelo programa:

- Intervalo de tempo entre a geração de processos (que irão para a fila de chegada)
- Tempo de uso da CPU por um processo
- Prioridade de uso do dispositivo de I/O (menor valor → maior prioridade)
- Intervalo de tempo entre solicitações de I/O
- Tempo de atendimento à operação de I/O

Os parâmetros anteriores devem ser exibidos ao longo da simulação.

Andamento da simulação:

O simulador irá usar o conceito de “tempo da simulação” através de eventos discretos. O que caracteriza um simulador por eventos discretos é o fato de o tempo da simulação ser descontínuo. O tempo anda aos saltos: suponha que um evento e_a ocorreu no instante t_a do tempo simulado e foi sucedido pelo evento e_b , o qual ocorreu no instante t_b do tempo simulado. Se não aconteceu nenhum evento de interesse entre e_a e e_b , então o tempo simulado deve pular de t_a diretamente para t_b . O tempo da simulação (contador global) deve ser apresentado à esquerda de cada linha e deve permitir ter uma idéia da evolução do sistema e de cada processo.

Com base nessa abordagem de tempo, o simulador deve exibir, para cada tempo simulado, um trace da execução onde aparece uma linha por alteração de estado dos processos. Estas linhas podem indicar que um novo processo começou ou retomou sua execução, que o processo corrente se bloqueou ou terminou. Quando não existe nenhum processo para executar, a CPU fica ociosa e essa situação deve ser representada pelo identificador de processo -1. Os demais processos serão identificados por um valor inteiro (contador).

O andamento da simulação deve ser exibido (trace de execução) na tela e salvo em um arquivo para posterior análise. O trace da execução dos processos deve exibir os estados pelos quais os processos passam durante seu ciclo de vida: *new, ready, running, blocked and terminated*. Periodicamente, ao longo do trace, deve ser exibida a situação atual das filas: fila de chegada, fila de prontos e fila de dispositivos.

Resultados da simulação:

O término da simulação se dá por meio do alcance do tempo total da simulação. Após a simulação, devem ser exibidos e salvos em um arquivo de saída as seguintes informações:

- Número de processos completados
- Numero de operações de I/O completadas
- Tempo de turnaround: por processo e média entre processos
- Tempo de espera: por processo e média entre processos
- Quantidade e percentual de tempo de CPU gasto na execução de processos
- Quantidade e percentual de tempo de CPU ociosa
- Quantidade e percentual de tempo total da simulação gasto em operações de I/O
- Para cada uma das 3 filas (chegada, prontos e dispositivos) calcular o tempo médio de espera e o tamanho médio da fila

Trabalho individual.

O que entregar:

- Implementação: todos os arquivo(s) fonte, com código comentado (funções comentadas).
- Análise: documento contendo uma análise comparativa de 2 experimentos totalmente distintos (que gerem filas de espera!!!) realizados com o simulador, contendo, para cada experimento:
 - Arquivo de entrada;
 - Arquivo de saída;
 - Análise crítica detalhada da experimentação.

Quando entregar:

- Prazo final: até o dia **08/11/2017**.
- A apresentação do trabalho será agendada posteriormente.

Como entregar:

- O arquivo **NomeDoAluno-escalona.c** deve ser enviado para pitthan@inf.ufsm.br e rwfazul@inf.ufsm.br, utilizando como Assunto da mensagem: [SO-escalona]-NomeDoAluno.

IMPORTANTE: O trabalho é composto por 2 etapas: implementação e análise. A avaliação irá considerar conjuntamente as 2 etapas (avaliação atômica). A não realização da etapa de Análise inviabiliza a avaliação do trabalho, uma vez que o mesmo torna-se um trabalho de Programação (e não de Sistemas Operacionais!!!).