INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CEARÁ

SISTEMAS OPERACIONAIS 2016.1

Prof. Fernando Parente Garcia

Projeto II – Estratégias de Alocação de Memória

Objetivo: Simular as estratégias de alocação de memória *first-fit*, *best-fit, worst-fit* e *next-fit.* As estratégias *first-fit*, *best-fit* e *worst-fit* foram abordadas em sala de aula. A estratégia *next-fit* inicia a busca a partir da posição da última alocação até encontrar o primeiro bloco que caiba o processo.

Descrição: O programa deve medir a utilização da memória (percentual de memória utilizada) e o tempo médio de espera dos processos. O tempo de espera de cada processo é o intervalo de tempo entre a criação e a conclusão do processo.

Entradas do Sistema:
Antes de iniciar a simulação, o usuário deverá configurar os seguintes parâmetros:

- Quantidade de processos que serão criados;
- Estratégia de alocação utilizada: *first fit, best fit, worst fit* ou *next-fit;*
- Tamanho da memória real (MB);
- Tamanho da área de memória ocupada pelo sistema operacional (MB);
- [$\mathbf{M_1}$, $\mathbf{M_2}$] - Intervalo para gerar aleatoriamente a área ocupada por cada processo na memória (MB);
- **[$\mathbf{t_{c1}}$, $\mathbf{t_{c2}}$]** - Intervalo para gerar aleatoriamente o tempo de criação de cada processo em relação ao processo criado anteriormente (segundos). Portanto, o instante de criação do processo é a soma do valor gerado aleatoriamente neste intervalo com o instante em que foi criado o processo anterior;
- [$\mathbf{t_{d1}}$, $\mathbf{t_{d2}}$] - Intervalo para gerar aleatoriamente a duração de cada processo (segundos). A duração representa quanto tempo o processo fica utilizando a memória a partir do instante em que ele foi alocado.

Saídas:
A cada instante, o programa deverá mostrar na tela as seguintes informações:

- Mapa da memória, mostrando a área de memória ocupada por cada processo e as áreas livres;
- Os processos que estão alocados na memória;
- Os processos que estão na fila aguardando a liberação de espaço de memória para serem alocados;
- Os instantes de tempo em que cada processo foi criado, foi alocado na memória e concluiu sua execução;
- Tempo de espera de cada processo (diferença entre o instante de conclusão e o instante de criação);

- Tempo médio de espera dos processos que já concluíram (média do tempo de espera de cada processo);
- Percentual de memória utilizada.

Funcionamento do programa:

Ao iniciar a execução, o programa deverá solicitar ao usuário a digitação de todos os dados de entrada. Em seguida, deverá gerar aleatoriamente o instante de criação, a duração e área de memória ocupada por cada processo.

Depois disto, o programa deverá iniciar a simulação alocando e liberando memória para os processos de acordo com seus parâmetros. Vale ressaltar que não deverá ser feito nenhum mecanismo de compactação, ou seja, um processo, ao ser criado, só deverá ser alocado na memória caso exista um buraco na memória que o caiba. Caso contrário, o processo deverá ficar na fila aguardando a liberação de um espaço de memória que o caiba.

Caso para teste e análise:

- Quantidade de processos que serão criados: **100**
- Tamanho da memória real (MB): **4000**
- Tamanho da área de memória ocupada pelo sistema operacional (MB): **500**
- [**50**, 4**00**] - Intervalo para gerar aleatoriamente a área ocupada por cada processo na memória (MB);
- **[10**, 3**0**] - Intervalo para gerar aleatoriamente o tempo de criação de cada processo em relação ao processo criado anteriormente (segundos). Portanto, o instante de criação do processo é a soma do valor gerado aleatoriamente neste intervalo com o instante em que foi criado o processo anterior;
- [**10**, **100**] - Intervalo para gerar aleatoriamente a duração de cada processo (segundos). A duração representa quanto tempo o processo fica utilizando a memória a partir do instante em que ele foi alocado.

Entregáveis:

Cada equipe deverá entregar, após a demonstração do trabalho, os seguintes artefatos:

- Arquivo compactado (ZIP) com o código fonte e o executável (JAR) do projeto;
- Relatório com os dados e análise do caso de teste descrito acima.

Data de entrega: 30/09/2016