

MAC0422 – Sistemas Operacionais – 2s2016

EP3

Data de entrega: 21/11/2016 até 8:00

Prof. Daniel Macêdo Batista

1 Problema

A tarefa neste EP é implementar um simulador de gerência de memória com diversos algoritmos para gerência do espaço livre e para substituição de páginas.

O simulador de gerência de memória deve receber como entrada um arquivo de trace, em texto puro, que possui como primeira linha:

```
total virtual s p
```

seguida de várias linhas com o seguinte formato:

```
t0 nome tf b p1 t1 p2 t2 p3 t3 [pn tn]
```

`total` é o total de memória física que deve ser simulada, `virtual` é o total de memória virtual que deve ser simulada e `s` é o tamanho da unidade de alocação a ser considerada para a execução dos algoritmos para gerência do espaço livre. `p` é o tamanho da página a ser considerada para a execução dos algoritmos de substituição de página. `t0` é o instante de tempo em segundos que um processo chega no sistema, `nome` é uma string sem espaços em branco que identifica o processo, `tf` é o instante de tempo no qual o processo é finalizado, `b` é a quantidade de memória utilizada pelo processo.

Os valores `p1, t1, ... pn, tn` dizem respeito às posições de memória, no espaço de endereço “local” do processo, acessadas pelo processo. `p1, t1` por exemplo informa que no instante de tempo `t1`, a posição `p1` é acessada pelo processo.

Todos os valores no arquivo de entrada são números inteiros.

O simulador deve finalizar sua execução assim que todos os processos do arquivo de entrada forem finalizados.

Com relação aos algoritmos para gerência do espaço livre, neste EP o simulador deve implementar os seguintes algoritmos, considerando que o controle de qual espaço está livre e qual está ocupado é feito usando bitmap:

1. *First Fit*
2. *Next Fit*
3. *Best Fit*
4. *Worst Fit*

Com relação aos algoritmos de substituição de página, neste EP o simulador deve implementar os seguintes algoritmos:

1. *Optimal* (nesse caso o arquivo de entrada deve ser lido antes para gerar os rótulos das páginas)
2. *Second-Chance*
3. *Clock*
4. *Least Recently Used* (Quarta versão)

2 Interação com o simulador

Quando executado na linha de comando (sem parâmetros) o simulador deve fornecer o prompt:

(ep2) :

Neste prompt os seguintes comandos precisam ser implementados:

- `carrega <arquivo>`: carrega o arquivo de nome `<arquivo>` para a simulação. Pode ser tanto o caminho relativo como absoluto do arquivo;
- `espaco <num>`: informa ao simulador que ele será executado com o algoritmo de gerenciamento de espaço livre de número `<num>`, de acordo com a numeração dos algoritmos apresentada anteriormente neste documento;
- `substitui <num>`: informa ao simulador que ele será executado com o algoritmo de substituição de páginas de número `<num>`, de acordo com a numeração dos algoritmos apresentada anteriormente neste documento;
- `executa <intervalo>`: executa o simulador e imprime o estado das memórias na tela de `<intervalo>` em `<intervalo>` segundos, juntamente com o conteúdo do bitmap que mantém o status da memória;
- `sai`: finaliza o simulador e volta para o shell do sistema operacional.

A memória deve ser simulada utilizando o arquivo `/tmp/ep2.mem` para a memória física e o arquivo `/tmp/ep2.vir` para a memória virtual. Estes arquivos devem ser criados toda vez que o simulador for inicializado e devem ter inicialmente um tamanho igual aos valores `total` e `virtual` definido no arquivo de entrada do simulador. Estes arquivos devem ser arquivos **binários** e devem conter inicialmente diversos valores -1 informando que toda a memória está livre para ser usada. À medida que a memória for sendo utilizada pelos processos simulados, as posições utilizadas por esses processos devem ser marcadas com números inteiros que identifiquem unicamente cada processo. Toda vez que um processo for carregado na memória ele deve escrever o número único que identifica ele (seria o equivalente ao PID no Linux) nas posições corretas dos arquivos. Toda vez que uma posição de memória for acessada, o PID também deve ser escrito nas posições corretas dos arquivos.

Em todos os momentos da simulação sempre haverá espaço na memória virtual suficiente para a soma dos tamanhos de todos os processos em execução nesses momentos.

3 Requisitos

O EP pode ser implementado em C, C++, Java, Python ou Ruby. A interação com o usuário deve ser feita inteiramente no prompt do EP, sem interface gráfica. Bibliotecas, funções, frameworks e similares que já implementem os algoritmos solicitados não podem ser usados. Caso sejam usados, a nota do EP será ZERO.

4 Sobre a entrega

Deve ser entregue um arquivo .tar.gz contendo os itens listados abaixo. EPs que não contenham **todos** os itens abaixo **exatamente como pedido** terão nota ZERO e não serão corrigidos. **A depender da qualidade do conteúdo entregue**, mesmo que o EP seja entregue, **ele pode ser considerado como não entregue, o que mudará o cálculo da média final**:

- código-fonte;
- arquivo LEIAME **em formato texto puro** explicando como compilar e executar o simulador;
- Makefile ou similar para facilitar a compilação do código-fonte;
- apresentação **em .pdf** para ser apresentada em no máximo 15 minutos resumindo os resultados obtidos com diversos experimentos e explicando detalhes da implementação dos algoritmos.

Os resultados devem ser exibidos com gráficos que facilitem observar o tempo necessário para encontrar um espaço livre na memória, comparando assim os quatro algoritmos para gerência do espaço livre, e a quantidade de *page faults*, comparando assim os quatro algoritmos de substituição de página. Considere 1 arquivo de trace suficiente para mostrar as diferenças entre os algoritmos. Como o tempo de execução pode ser diferente em várias execuções dos algoritmos, apresente os resultados obtidos com gráficos em barra. Cada valor a ser apresentado nos gráficos deve possuir média e intervalo de confiança de 30 medições com nível de 95%. Os resultados dos experimentos valem 3,0 pontos.

O desempacotamento do arquivo .tar.gz deve produzir um diretório contendo os itens. O nome do diretório deve ser ep3-membros_da_equipe. Por exemplo: ep3-joao-maria. EPs que não gerem um diretório ou que gerem o diretório com o nome errado perderão 1,0 ponto.

A entrega do .tar.gz deve ser feita através do PACA.

O EP pode ser feito individualmente ou em dupla.

Obs.: não inclua no .tar.gz itens que não foram pedidos neste enunciado. Relatórios, saídas para diversas execuções e entradas usadas para testar o programa não devem ser entregues. No máximo um resumo sobre as entradas usadas pode ser colocado na apresentação, caso isso não ocupe muito espaço.