

Trabalho de Sistemas Operacionais I

Simulador de gerência de memória utilizando a política LRU com a variação segunda-chance

Questionário:

1. Nome dos componentes do grupo e número do cartão.

Eduardo Mattos da Silva Mello - 141964

João Luiz Grave Gross - 180171

2. Descrição da plataforma utilizada para desenvolvimento. Qual o tipo de processador (número de cores, com ou sem suporte HT)? Qual a distribuição GNU/Linux utilizada e a versão do núcleo? Qual a versão do gcc? Se o trabalho foi feito ou não em ambientes virtualizados? Em caso afirmativo, qual a máquina virtual utilizada (versão)?

Processador: Intel Core 2 Duo CPU E7400 @ 2.80GHz, sem suporte a HT.

Distribuição GNU/Linux: Ubuntu 11.04 e Gnome Versão: 2.32.1

Ambiente virtualizado: Não

gcc: versão 4.5.2-1ubuntu3.

3. Descrever a estrutura de dados empregada para encadear as páginas no LRU para permitir a análise delas como se fosse o movimento de um ponteiro de relógio (algoritmo segunda chance).

Para o algoritmo segunda-chance foi utilizada uma lista do tipo FIFO, onde cada elemento da lista contém um inteiro para armazenar o número de um frame da memória que possui uma página e um inteiro para armazenar o bit de referência.

4. Descrever a estrutura da tabela de páginas.

Para a gerência dos processos criamos uma lista do tipo fifo. Quando um processo é criado ele é adicionado na lista e quando encerrado é excluído. Cada processo possui uma lista com as suas páginas, onde cada página possui um número de 1 a n, onde n é o número de páginas do processo, e outras 3 variáveis que armazenam os acessos e swaps feitos sobre essa página, ou seja, número de acessos, número de substituições e número de page faults. Logo, a tabela de páginas é implementada em uma lista.

5. Explicitar, em função dos bits de referência e modificação, qual foi a ordem de preferência para escolher uma página vítima (modificada e acessada; acessada e não modificada; não acessada e modificada; não acessada e não modificada). Explique o porque, as vantagens e as desvantagens dessa escolha. Utilize, também, os seus casos de teste para justificar a resposta.

A versão do algoritmo segunda chance implementada foi a versão padrão que considera apenas o bit de referência. Adotou-se tal escolha, por ser mais simples e por não haver exigência na definição do simulador de gerência de memória para implementar o algoritmo segunda chance aprimorado, que contém o bit de modificação (de acordo com SILBERSCHATZ, edição 7, seção 9.4.5.3, página 337). Logo, a escolha

da página vítima se deu com base nesse algoritmo, ou seja, varrendo a fifo circular até encontrar um elemento com bit de referência igual a zero.

Assim, toda a página escolhido como vítima é copiado para a área de swap e depois a nova página é gravada no mesmo frame da página vítima. Para fins práticos esta implementação não é muito eficiente, pois haverá casos em que a área de swap será atualizada sem que haja necessidade, quando, por exemplo, a página vítima alocada no frame não houver sido alterada. Como vantagem podemos citar a fácil compreensão do algoritmo, estruturas com menores, bem como a simplicidade na escolha de uma página vítima.

6. Descrever o que funciona e o que NÃO está funcionando no simulador desenvolvido. Em caso de não funcionamento, dizer qual é a sua visão do porquê deste não funcionamento.

As estruturas de memória, processos, e fifo circular estão funcionando, bem como as demais operações solicitadas na descrição do trabalho com a implementação do parser, leitura do arquivo e geração do log, e término do programa por erros no arquivo de configuração.

7. Qual a metodologia utilizada para testar o simulador? Isto é, quais foram os passos (e arquivos de teste) efetuados para testar o simulador? Foi utilizado um debugger? Qual?

Para debugar o programa utilizamos muita análise de conteúdo de variáveis com printf e com breakpoints (getchar). Mas também utilizamos o GDB, que é um debugger que nos possibilita ver o que ocorre com o conteúdo das variáveis em tempo de execução ou mesmo o que aconteceu quando ocorre uma falha de segmentação (se alocamos um ponteiro e não desalocamos, acessos a trechos de memória que não pertencem ao nosso código, etc).

8. Quais as principais dificuldades encontradas e quais as soluções empregadas para contorná-las.

A maior dificuldade ocorreu ao implementar o algoritmo segunda chance, pois era necessário atualizar o conteúdo da memória, processos e lista LRU. No caso de substituições de páginas, atualizar as informações de acessos das páginas na lista de processos e também a fila circular de frames do algoritmo segunda chance, no que se refere à atualização dos nodos da lista e escolha dos frames vítimas para a substituição.