

Relatório do trabalho sobre Gerência de Memória com Segmentação e Memória Virtual

Grupo: Bruno Behnken, Hugo Faria, Lucas Murakami, Rafael Mendes, Thales Nathan

Estratégia de Alocação de Segmentos:

Nosso trabalho utiliza estratégia First-fit para alocação de segmentos na Memória Principal. Para guardar os espaços livres, é utilizada uma lista encadeada.

Estratégia de Realocação de Segmentos:

Para a estratégia de Realocação, usamos o algoritmo LRU. Para isso, temos uma estrutura de dados que guarda os segmentos atuais na Memória Principal onde são guardadas a "idade" de cada segmento, renovada a cada vez que este é referenciado.

Como funciona a simulação:

Para iniciar o programa use o comando `./principal Memoria`, onde "Memoria" será um inteiro que representará o tamanho total da Memória Principal na simulação.

Ao iniciar o programa, um menu informa ao usuário as opções de ação que são:

1 – Inserir Processo no Sistema:

Inclui um novo processo no sistema, que passará a ter seus segmentos passíveis de alocação em M.P.

2 – Excluir Processo do Sistema:

Exclui um processo do sistema, de forma que seus segmentos já não poderão mais ser acessados.

3 – Acessar Endereço Real:

É informado ao sistema qual o PID do processo que tem o segmento que se deseja acessar e o Endereço Virtual que se deseja acessar (o número desse segmento na tabela de segmentos e o offset).

Após passar essas informações, o sistema aloca, se necessário, o segmento na M.P. e atualiza as "idades" de cada segmento alocado de acordo com a estratégia do LRU. O sistema também mantém atualizadas as tabelas de segmentos envolvidas nas possíveis realocações, setando bits de presença e as bases em M.P. de cada segmento alocado.

Além de guardar bits de presença e base em M.P., a tabela de segmentos também guarda o tamanho de cada segmento, para diversos fins como checar se o offset dado é válido (dentro dos limites do segmento).

Finalmente, é exibido para o usuário o endereço real em M.P. da base do segmento desejado assim como a mapeação do endereço virtual para real fazendo "base + offset".

4 – Ver os Intervalos de Espaço Livre na M.P.:

Mostra a lista encadeada de espaços livres na M.P.

Ex: (300,500) → (600, 700), representando que na M.P. existem espaços livres

entre as posições de memória 300 e 500 assim como entre 600 e 700.

5 – Ver os Processos e seus Segmentos:

Mostra informações dos processos que estão incluídos no sistema no momento.

0 – Encerrar Programa:

Encerra simulação.

Problemas:

Para algumas ordens de alocação de segmentos, o sistema apresenta um funcionamento incorreto do cálculo de espaços livres. Podendo gerar representações de espaços livres erradas e consequentemente existindo a possibilidade de alocações erradas. Ainda não conseguimos reparar esses casos.

Exemplo de erro:

- Inicie o programa com uma M.P. de tamanho 1000.
- Inclua 2 processos no sistema. Um com PID = 1 e apenas 1 segmento de tamanho 500. O outro com PID = 2 e 2 segmentos, um de tamanho 500 e o outro 300.
- Acesse o único segmento do processo 1 com um offset qualquer.
- Acesse o segmento de tamanho 300 do processo 2 com offset qualquer.

Perceba que até este momento o funcionamento do sistema foi correto! Bits de presença setados de acordo com a alocação de segmentos em M.P, endereços virtuais corretamente mapeados para os endereços reais. Também é possível utilizar a opção 4 ("Ver os Intervalos de Espaço Livre na M.P.") antes, entre e depois de cada acesso a segmento feito para checar que a lista de espaços livres se mantém correta.

- Agora, acesse o segmento de tamanho 500 do processo 2 com offset qualquer.

Após esse acesso, veja que embora os endereços reais calculados e os bits de presença (que podem ser checados novamente com a opção 5) estejam corretos. A opção 4 nos mostra uma Lista de Espaços Livres incorreta, com um intervalo "(500,500)", que não deveria estar lá, até porque isso não representa um intervalo de espaço livre.

- Agora, acesse novamente o único segmento do processo 1.

E então temos uma falha de segmentação...

Conclusão:

Chegamos a um programa que é capaz de simular um gerenciamento de memória com segmentação e memória virtual. Porém, como alguns casos de uso mostram, este ainda contém alguns erros não solucionados.