

AULA 15 - Algoritmos de Substituição de Páginas

Os algoritmos de substituição de páginas são políticas definidas para escolher qual(is) página(s) da memória dará lugar a página que foi solicitada e que precisa ser carregada. Isso é necessário quando não há espaço disponível para armazenar a nova página. Devemos ressaltar que se a página a ser removida sofreu alterações enquanto esteve na memória, a cópia virtual existente em disco deverá ser atualizada. Por outro lado, se a página não foi modificada significa que sua cópia está atualizada e, portanto, não é necessário reescrevê-la. Políticas de substituição de páginas devem ser utilizadas em sistemas que fazem uso de memória virtual paginada com o objetivo de melhorar o desempenho do sistema computacional.

Os algoritmos de substituição de páginas podem ser classificados, basicamente, em: algoritmos com espaço fixo e algoritmos com espaço variável. A diferença entre estes dois tipos de algoritmos é que o de espaço fixo trabalha sobre uma área de memória sempre constante, enquanto que os de espaço variável podem modificar o tamanho da memória alocada dinamicamente [CAS03].

Algoritmo de substituição de páginas FIFO

O FIFO (*First-in, First-out*) é um algoritmo de substituição de páginas de baixo custo e de fácil implementação que consiste em substituir a página que foi carregada há mais tempo na memória (*a primeira página a entrar é a primeira a sair*). Esta escolha não leva em consideração se a página está sendo muito utilizada ou não, o que não é muito adequado pois pode prejudicar o desempenho do sistema. Por este motivo, o FIFO apresenta uma deficiência denominada *anomalia de Belady*: a quantidade de falta de páginas pode aumentar quando o tamanho da memória também aumenta. Por estas razões, o algoritmo FIFO puro é muito pouco utilizado. Contudo, sua principal vantagem é a facilidade de implementação: uma lista de páginas ordenada pela “idade”. Dessa forma, na ocorrência de uma falta de página a primeira página da lista será substituída e a nova será acrescentada ao final da lista.

Algoritmo de substituição de páginas LRU

O LRU (*Least Recently Used*) é um algoritmo de substituição de página que apresenta um bom desempenho substituindo a página *menos recentemente usada*. Esta política foi definida baseada na seguinte observação: se a página está sendo intensamente referenciada pelas instruções é muito provável que ela seja novamente referenciada pelas instruções seguintes e, de modo oposto, aquelas que não foram acessadas nas últimas instruções também é provável que não sejam acessadas nas próximas. Apesar de o LRU apresentar um bom desempenho ele também possui

algumas deficiências [CAS03] quando o padrão de acesso é sequencial (em estruturas de dados do tipo vetor, lista, árvore), dentro de loops, etc. Diante dessas deficiências foram propostas algumas variações do LRU, dentre eles destacamos o LRU-K. Este algoritmo não substitui aquela que foi referenciada há mais tempo e sim quando ocorreu seu *k-último* acesso. Por exemplo, LRU-2 substituirá a página que teve seu penúltimo acesso feito há mais tempo e LRU-3 observará o antepenúltimo e assim por diante.

A implementação do LRU também pode ser feita através de uma lista, mantendo as páginas mais referenciadas no início (cabeça) e a menos referenciadas no final da lista. Portanto, ao substituir retira-se a página que está no final da lista. O maior problema com esta organização é que a lista deve ser atualizada a cada nova referência efetuada sobre as páginas, o que torna alto o custo dessa manutenção.

Algoritmo de substituição de páginas Ótimo

O algoritmo ótimo, proposto por Belady em 1966, é o que apresenta o melhor desempenho computacional e o que minimiza o número de faltas de páginas. No entanto, sua implementação é praticamente impossível. A idéia do algoritmo é retirar da memória a página que vai demorar mais tempo para ser referenciada novamente. Para isso, o algoritmo precisaria saber, antecipadamente, todos os acessos à memória realizados pela aplicação, o que é impossível em um caso real. Por estes motivos, o algoritmo ótimo só é utilizado em simulações para se estabelecer o valor ótimo e analisar a eficiência de outras propostas elaboradas.

Algoritmo de substituição de páginas MRU

O algoritmo MRU (*Most Recently Used*) faz a substituição da última página acessada. Este algoritmo também apresenta algumas variações, semelhante ao LRU. Por exemplo, o MRU-n escolhe a n-última página acessada para ser substituída. Dessa forma, é possível explorar com mais eficiência o princípio de localidade temporal apresentada pelos acessos.

Algoritmo de substituição de páginas CLOCK

Este algoritmo mantém todas as páginas em uma lista circular (em forma de relógio). A ordem mantida segue a seqüência em que elas foram carregadas em memória. Além disso, é adicionado um bit de uso que indica se a página foi referenciada novamente depois de ter sido carregada. Ao precisar substituir uma página o algoritmo verifica se a página mais antiga está com o bit zerado (o que significa que a página não foi mais referenciada) para ser substituída. Se ela não estiver o bit é zerado e a próxima página da fila mais antiga será verificada. Esse processo continua até que uma página antiga com o bit zerado seja encontrada para ser substituída.

Algoritmo de substituição de páginas NRU

O algoritmo NRU (*Not Recently Used*) procura por páginas que não foram referenciadas nos últimos acessos para serem substituídas. Tal informação é mantida através de um bit. Este algoritmo também verifica, através de um bit de modificação, se a página teve seu conteúdo alterado durante sua permanência em memória. Esta informação também vai ajudar a direcionar a escolha da página. As substituições das páginas seguem a seguinte prioridade: páginas não referenciadas e não modificadas, páginas não referenciadas, páginas não modificadas e páginas referenciadas e modificadas.

Algoritmo de substituição de páginas LFU

O LFU (*Least Frequently Used*) escolhe a página que foi menos acessada dentre todas as que estão carregadas em memória. Para isso, é mantido um contador de acessos associado a cada página (*hit*) para que se possa realizar esta verificação. Esta informação é zerada cada vez que a página deixa a memória. Portanto, o problema desse algoritmo é que ele prejudica as páginas recém-carregadas, uma vez que por estarem com o contador de acessos zerado a probabilidade de serem substituídas é maior. Qual uma possível solução para este problema? (Estabelecer um tempo de carência) Só páginas fora desse tempo é que podem ser substituídas. Tal estratégia deu origem ao algoritmo FBR (*Frequency-Based Replacement*).

Algoritmo de substituição de páginas MFU

O MFU (*Most Frequently Used*) substitui a página que tem sido mais referenciada, portanto, o oposto do LFU. O controle também é feito através de contadores de acesso. O maior problema deste algoritmo é que ele ignora o *princípio de localidade temporal*.

Algoritmo de substituição de páginas WS

O algoritmo WS (*Working Set*) possui a mesma política do LRU. No entanto, este algoritmo não realiza apenas a substituição de páginas ele também estabelece um tempo máximo que cada página pode permanecer ativa na memória. Assim, toda página que tem seu tempo de permanência esgotado ela é retirada da memória. Portanto, o número de páginas ativas é variável. O WS assegura que as páginas pertencentes ao *working set* processo permanecerão ativas em memória. Os algoritmos apresentados são alguns dos disponíveis na literatura. Outras implementações ou variações dos destacados aqui podem ser encontradas também [CAS03].

6. Explique como funciona as políticas: FIFO, ÓTIMA, LRU e LFU.

Bibliografia

TANENBAUM, A. S. *Sistemas Operacionais Modernos*. Prentice Hall, 2ª Edição, 2003. (Cap. 4)

Referência Bibliográfica

[CAS03] CASSETTARI, H. H. Análise da Localidade de Programas e Desenvolvimento de Algoritmos Adaptativos para Substituição de Páginas. Qualificação de Mestrado. Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais. 2003.