

Gerenciamento de Memória

No campo da tecnologia da informação, o gerenciamento de memória é um tema que tem muita importância, uma atenção especial é voltada a ele, várias técnicas são constantemente desenvolvidas para a sua melhora. Os programadores necessitam de mais memória a cada dia, a necessidade de ter vários programas sendo executados ao mesmo tempo também aumenta, e com esse aumento vários requisitos podem perder qualidade, como o desempenho. A memória de um computador não é infinita, e quando a memória não tem mais espaço disponível ocorrem erros.

O gerenciamento de memória é essencialmente formado pela alocação e reciclagem da memória:

1. Alocação da Memória:

Acontece quando o programa, que está sendo executado, requisita um bloco de memória, o gerenciador por sua vez, disponibiliza o bloco de memória para a alocação desejada.

1.1 Alocação Estática:

Bloco de memória alocado quando o programa é compilado, comandado pelo sistema operacional, que ao ler o programa, reserva o bloco de memória desejado.

1.2 Alocação Dinâmica:

A alocação é feita quando o programa é executado, a memória reservada para a alocação dinâmica é chamada de “heap”, em linguagens que não tem o gerenciamento automático de memória, como a linguagem C, existem vários métodos de organizar o heap. Em outras linguagens o gerenciamento é automático, como na linguagem Java, esse gerenciamento automático é chamado de Garbage Collector (Coletor de lixo), assunto que será tratado posteriormente.

2. Reciclagem da Memória:

Depois que um bloco da memória é alocado, e não será mais utilizado, não tendo mais nenhuma referência a ele pelo programa, esse bloco pode então ser liberado, sendo liberado poderá ser utilizado novamente, ficando assim livre e a disposição novamente do programa que está sendo executado.

2.1 Reciclagem Interna:

Quando os blocos de memória, chamados de página, não são ocupados por completo pelo processo a eles destinado, é gerado assim um espaço que não está sendo utilizado e poderá, se liberado, ser utilizado novamente.

2.2 Reciclagem Externa:

Quando os programas vão sendo finalizados, deixam lacunas entre as páginas criadas, se essas lacunas forem muito pequenas, o suficiente para que o que precisa ser escrito na memória não caiba nela, essas lacunas ficam inutilizadas.

A paginação é um processo virtual da memória, que divide a memória física em “frames” (pequenas partições), permitindo assim, uma utilização mais eficiente da mesma. Quando uma alocação de memória é solicitada, são páginas que são liberadas, que é a menor unidade desse processo. Cada página é mapeada em uma frame, chamado assim de paginação.

Gerenciamento Manual de Memória

O programador é responsável de especificar explicitamente, quando e como a memória será alocada, mudando assim a configuração do programa de acordo com a sua necessidade e (ou) vontade. Nesse modelo de gerenciamento, problemas conhecidos como “vazamentos de memória” podem ocorrer, tais problemas ocorrem quando um bloco da memória é alocado, e mesmo que não seja mais útil, não é liberado, mantendo assim o seu uso, desperdiçando um espaço que poderia ser utilizado por outro processo, que esteja sendo executado.

Gerenciamento Automático

Conhecido como Garbage Collector, libera os blocos da memória que não estão mais sendo utilizados automaticamente. O bom conhecimento sobre esse tema é muito importante para desenvolver programas com alto desempenho, pois se um programa forçar o sistema operacional a usar a memória virtual, seu desempenho não será muito eficiente.

O Garbage Collector libera o programador de ter que gerenciar a memória do programa, não precisando assim, o programador, de ficar atento aos detalhes do gerenciamento de memória, pois ele já faz isso de forma automática, reduzindo ou até eliminando possíveis defeitos dos softwares desenvolvidos, se tratando de uma técnica mais confiável. A região da memória pode ser liberada mais de uma vez, diminuindo assim a possibilidade de ocorrer um esgotamento de memória.

Para decidir quais blocos da memória podem ou não ser liberados, o Garbage Collector usa processos que consomem recursos computacionais, diminuindo assim o desempenho do programa, devido à sobrecarga do seu uso. E também deixa o programador desatento aos detalhes de memória.

A liberação automática de memória ocorre através de algoritmos. Existem várias estratégias para isso, que dependendo do programa criado podem ser vantajosos ou desvantajosos, apresentando impacto direto no desempenho. O objetivo desses algoritmos é garantir a identificação precisa de todos os ponteiros e a reciclagem de toda a memória inutilizada.

Dentre os algoritmos de gerenciamento de memória automática, podemos citar como os mais conhecidos:

- Reference counting algorithm (Collins 1960);
- Cycle collecting algorithm (Bobrow 1980);
- Mark and sweep algorithm (McCarthy 1960);
- Copying algorithm (Cheney 1970).

Referências:

PINTO, Flávio; LUNA, Hedley. Gerenciamento de Memória em Java.
<http://www.devmedia.com.br/desmistificando-o-garbage-collector/5451>
<http://pt.scribd.com/doc/73307165/25/Gerenciamento-manual-de-memoria>