

Gerência de memória no Linux

1 Memória no Linux

O Linux é um sistema operacional com memória virtual paginada, isto quer dizer que podemos ter programas em execução cujo tamanho é maior que a memória física disponível para executá-los.

Assim, o Linux é responsável por manter as partes dos programas/códigos em uso na memória, deixando o resto no disco rígido, só os carregando a partir do momento que são solicitados a serem usados.

Por exemplo, um programa de 40 MB pode ser executado pelo processador em uma página de 4 MB de memória, já que o Linux carrega os 4 MB do software que serão executados, só carregando as partes que serão usados para a memória a partir do momento que são solicitados para executar pelo usuário.

No Linux, os processos que estão em execução no processador têm prioridade para estarem na memória, no entanto, quando terminam de executar, havendo espaço livre (free) na memória, para posterior execução no processador caso sejam requisitados novamente, economizando, assim leitura e escrita em disco.

Assim, os dados não são apagados imediatamente após o encerramento da sua execução pelo processador, mas apenas marcados como não executando e, portanto, passíveis de se sobrescrever por outro código caso haja necessidade de memória. Esse procedimento garante maior agilidade na execução dos processos, pois, como dito anteriormente, há economia de leitura e gravação no disco.

No entanto, em caso de haver pouca memória no computador e a memória RAM esteja cheia de códigos que estejam sendo executados no processador, o Linux utiliza a memória SWAP, que é a área do disco reservado justamente para funcionar como uma memória provisória virtual (swap = troca).

A área de troca, swap, no Linux, é considerada tão importante que se cria uma partição específica para essa troca.

2 Chamada ao sistema

O espaço de endereçamento na memória é criado quando o código a ser executado no processador é criado e sobrescrito através de uma chamada ao sistema (exec).

As chamadas ao sistema mais importantes são:

- fork: cria um novo processo;
- wait: processo-pai aguarda o término de seu processo-filho;
- exec: executa outro programa;
- exit: terminar sua execução;

3 Buffer/Cache

O kernel do linux é desenvolvido de forma que considera que toda memória livre trata-se de um desperdício.

Neste contexto, o kernel Linux utiliza a memória que estiver livre (free) para ajudar outro subsistema que tem como atribuição: a memória de armazenamento.

O acesso ao disco, em qualquer sistema operacional, tanto para leitura quanto para escrita, é muito mais lento que o acesso à memória RAM.

Dessa forma, o kernel do linux, já desenvolvido sabendo-se disso, armazena na memória RAM todo o dado que é acessado no disco, aproveitando-se para tal da memória livre.

Assim, se o processador precisar consultar esse mesmo dado outra vez, ele consulta o dado que está na memória RAM, que tem o acesso muito mais rápido.

Dessa forma, o acesso aos dados, à partir da segunda vez, serão muito mais rápidos, já que no primeiro acesso estão no disco e tem de ser carregados para a memória, mas que a partir do segundo acesso já estarão na memória RAM.

Esse exemplo pode ser ilustrado usando o comando free como a seguir:

```
$ free -m
```

| | total | used | free | shared | buff/cache | available |
|-------|-------|------|------|--------|------------|-----------|
| Mem: | 7755 | 4404 | 1085 | 1305 | 3875 | 3350 |
| Swap: | 1023 | 38 | 985 | | | |

A saída do comando acima mostra que há 3493 MB de dados provenientes do disco em cache.

Quanto ao Buffer, trata-se de dados que são uma espécie de ponteiro para a posição dos arquivos nos discos.

Assim, o uso do comando “ls” para listar arquivos em uma pasta/diretório diz ao kernel para verificar a existência dos arquivos no disco e armazenar a posição deles na memória, gerando assim o “buffer”.

Apesar dos termos “buffer” e “cache” serem diferentes, na prática, todos usam os termos com a mesma definição, a de memórias auxiliares à RAM, acelerando o acesso aos dados no disco.

O Cache pode apresentar dois conceitos diferentes no Linux:

- cache limpo: os dados que estão nesse cache e que não foram alterados, portanto, são idênticos aos dados no disco, normalmente, a maior parte do cache é limpo;

- cache sujo: são os dados que foram alterados no cache e que, por isso, são diferentes dos que estão no disco.

Quando a execução de um processo “suja” um dado no cache, esse dado precisa ser atualizado no disco, portanto, precisando ser gravado. Nesse caso, o esforço computacional para se liberar esse cache é um pouco mais custoso e não seria totalmente correto afirmarmos que esses dados podem ser considerados livres, mas normalmente é a menor parte do cache que está nessa situação.

Por isso não devemos ver o cache como memória não disponível, salvo algumas exceções, como alguns servidores de banco de dados que geram muito cache sujo por fazerem muitas operações de escrita.

Apesar das exceções, a regra geral é considerar o buffer/cache como memória disponível para os processos.

```
$ free -m
```

| | total | used | free | shared | buff/cache | available |
|-------|-------|------|------|--------|------------|-----------|
| Mem: | 7755 | 4404 | 1085 | 1305 | 3875 | 3350 |
| Swap: | 1023 | 38 | 985 | | | |

A segunda linha da saída do comando acima exibe a quantidade de memória em uso (used) e a quantidade livre (free) considerando o cache como memória disponível. Portanto, é ela que deve ser utilizada para julgarmos quanta memória está em uso e quanta memória está disponível.

4 Swap

o “swap”, também chamado de memória de troca ou memória virtual, trata-se de uma área em disco à qual o sistema recorre quando não há mais memória

disponível e novos processos estão surgindo, precisando de memória para serem alocados.

Um processo ativo precisa estar na memória RAM para ser executado.

O que o sistema operacional faz então, é mover alguns dados de alguns processos, que não estão sendo muito utilizados, para a área de swap.

Isso libera espaço na memória RAM e é nesse espaço que o processo será executado, pois não se executa um processo do swap.

Se até mesmo o swap estiver lotado, o kernel entrará em um modo chamado de OOM (Out of Memory), onde ele irá eliminar processos na tentativa desesperada de liberar espaço.

Assim, o computador deixaria de responder por um tempo, no entanto, com o passar do tempo, algum processo será morto pelo kernel e será possível interagir com o servidor novamente.