

Infraestrutura de Software

Questão Aberta 2^a Unidade

Leonardo Medeiros de Freitas¹

¹Ciência da Computação – CESAR School
Avenida, Cais do Apolo, 77, Recife - PE, 50030-22

lmf@cesar.school

Abstract. *The objective of the document is mainly to provide information about virtual memory, memory fragmentation related to virtual memory, and memory space allocation related to three different operating systems: Linux, Windows, and MacOS. In addition, the document also includes various explanations and definitions of important concepts for understanding the aforementioned topics.*

Keywords — *memory, kernel, management, fragmentation, Linux, Windows, MacOS*

Resumo. *O objetivo do documento é principalmente informar sobre memória virtual, fragmentação relacionada à memória virtual e alocação do espaço de memória relacionados a três sistemas operacionais diferentes, sendo eles o SO Linux, o SO Windows e o SO MacOS. Além disso, o documento também tem diversas explicações e definições de conceitos importantes para o entendimento dos tópicos já citados.*

Palavras-chave — *memória, kernel, gerenciamento, fragmentação, Linux, Windows, MacOS*

1. Introdução

Para entender o conceito de fragmentação de memória e memória virtual num geral, é importante a explicação destes conceitos. Resumidamente, memória virtual é um método e conceito utilizado em diversos SO(Sistemas Operacionais) para gerenciar a memória de um computador e é uma extensão da memória física, que usa memória secundária para compensar a escassez de memória física, e permite que o sistema utilize parte do disco como uma área temporária para guardar dados e informações que não estão sendo utilizados no momento pela CPU. A memória virtual também permite que os processos compartilhem arquivos e implementem a memória compartilhada. Além disso, ela fornece um mecanismo eficiente para a criação de processos. Para se entender a memória virtual, o conceito de endereço virtual é de extrema importância. Resumidamente, um endereço virtual é um número binário na memória virtual que permite que um processo use um local no armazenamento primário (memória principal) ou, em alguns casos, no armazenamento secundário. (SILBERSCHATZ; GALVIN; GAGNE, 2015)

Para a introdução da fragmentação no contexto de memória virtual, a explicação do que são frames e páginas no contexto de software é de grande importância. Deste modo, pagina(page) é uma unidade de alocação de memória que possui um endereço virtual. Quando programas são executados, as instruções e dados são divididos em páginas

virtuais. O sistema operacional é o responsável por organizar e gerenciar o mapeamento, para que as páginas sejam acessadas conforme o necessário.(DEITEL; DEITEL; CHOFFNES, 2005)

Já os frames se referem a unidades de blocos de memória física que correspondem ao tamanho das páginas na memória virtual. Esses frames são utilizados para o armazenamento das páginas virtuais mapeadas pela própria memória virtual. Os frames são alocados dinamicamente conforme a necessidade para a execução dos processos. O sistema operacional garante que a memória física seja utilizada de maneira eficiente e que os processos tenham acesso às páginas que forem precisas para a sua execução.(DEITEL; DEITEL; CHOFFNES, 2005)

Em resumo, páginas e frames se complementam, pois as páginas são unidades de memória virtual e frames são blocos que correspondem à memória física usados para guardar as páginas, e são utilizados para que a execução dos programas seja mais eficiente e que o compartilhamento da memória entre processos seja mais seguro.(DEITEL; DEITEL; CHOFFNES, 2005)

Em relação a fragmentação de memória virtual, ela ocorre quando o espaço disponível na memória virtual de um Sistema Operacional é dividido em blocos não adjacentes, o que dificulta a alocação com eficiência da memória para a realização dos processos. Essa fragmentação leva a problemas de desempenho, limita a capacidade do sistema, diminui a eficiência da memória cache e aumenta a quantidade de vezes que a operação de troca entre a memória física e o disco rígido acontecem.(DEITEL; DEITEL; CHOFFNES, 2005)

Existem dois tipos diferentes de fragmentação de memória virtual. Um deles é a fragmentação interna, que ocorre quando um processo reserva para si uma parte da memória maior do que realmente é necessário. Isso acontece pois os sistemas operacionais normalmente alocam memória em unidades fixas, assim como as páginas de tamanho fixo. Se um processo pedir uma quantidade de memória menor do que o tamanho da unidade alocada, o resto do espaço é inutilizado por outros processos, e consequentemente, desperdiçado. Isso resulta em fragmentação interna, onde há lacunas de espaço não utilizadas dentro dos blocos alocados.(SILBERSCHATZ; GALVIN; GAGNE, 2015)

E existe a fragmentação externa, que ocorre quando existem espaços vazios nas lacunas da memória virtual, mas essas lacunas não estão próximas das outras, não estão contíguas. Ao passo que os processos são alocados e deslocados, é possível que a memória fique dividida em lacunas, não possibilitando a alocação de blocos maiores para novos processos. Essa fragmentação externa leva à utilização ineficiente da memória.(SILBERSCHATZ; GALVIN; GAGNE, 2015)

Por fim, a alocação de memória é o processo de reservar espaço virtual ou físico no computador para uma finalidade específica (por exemplo, para a execução de programas e serviços). A alocação de memória faz parte do gerenciamento dos recursos de memória do computador. Por meio da alocação de memória, os programas e serviços de computador recebem uma porção de memória específica, dependendo de quanta memória precisarem.(MICROSOFT. . . , 2021a)

Em diversas linguagens de programação existem comandos que servem para alocar memória dinamicamente. Na linguagem de programação C, os comandos que fazem

tal ação são o “malloc()” e o “calloc()”.(MICROSOFT... , 2021b)

2. Linux

Durante o desenvolvimento do kernel do Linux, houve diversas melhorias no gerenciamento de memória com o objetivo de aumentar a escalabilidade e o desempenho do sistema. Em relação a organização da memória, o Linux aloca a sua memória com páginas de tamanho único, comumente quatro kilobytes(kb). O kernel armazena as informações sobre cada frame em uma estrutura de página. No que diz respeito à organização da memória virtual no SO Linux, em um sistema de 32 bits, cada processo pode endereçar 2^{32} bytes, o que significa que cada endereço de memória virtual tem 4 gigabytes. Em um sistema de 64 bits, o kernel suporta endereços de memória virtual de tamanhos maiores. O tamanho do endereço depende da configuração do sistema, mas pode chegar à casa dos petabytes(PB) de memória.(DEITEL; DEITEL; CHOFFNES, 2005)

Embora o espaço de endereço virtual de um processo seja composto de páginas individuais, o kernel usa um mecanismo chamado de área de memória virtual, para organizar a parte da memória virtual que o processo está usando. Uma área de memória virtual descreve um conjunto adjacente de páginas no espaço de endereço virtual de um processo. O kernel pode armazenar um código executável do processo, stack e heap na memória em áreas separadas e distintas de memória virtual.(DEITEL; DEITEL; CHOFFNES, 2005)

Quando um processo pede memória adicional, o kernel faz uma tentativa para satisfazer o pedido, aumentando uma área já existente da memória virtual. O kernel fornece um alocador adicional para que essas áreas pudessem ser alocadas, com a função kmalloc() (semelhante a função malloc() e calloc(), muito utilizada na linguagem C), que aloca páginas inteiras sob demanda, mas depois as divide em partes menores. O kernel mantém diversas listas de páginas que estão em uso pela função kmalloc(). A área da memória virtual que o kernel seleciona depende do tipo de memória que o processo está solicitando e para que vai ser usado (como por exemplo os já citados stack e heap). Se o processo solicitar memória que não corresponde a uma área de memória virtual existente, ou se o kernel não puder alocar um espaço de endereço adjacente do tamanho que foi pedido em uma área de memória virtual existente, o kernel cria uma nova área de memória virtual.(DEITEL; DEITEL; CHOFFNES, 2005)

O kernel do Linux é responsável por manter o espaço de endereçamento acessível aos processos e criações de páginas a partir do disco e armazenamento, quando for preciso. Deste modo, o kernel conta com uma área de memória virtual dividida em duas partes. A primeira parte conta com o núcleo do kernel juntamente com as páginas alocadas. Ademais, é uma região estática que conta com as referências da tabela de páginas disponíveis na memória física do sistema. Contudo, a segunda região da área de memória do kernel do SO Linux não é destinada para fins específicos, podendo ser usada de outras maneiras.(SILBERSCHATZ; GALVIN; GAGNE, 2015)

A fragmentação da memória virtual do linux se dá pelo swapping, que é basicamente o gerenciador de memória do linux determinando quais páginas é preciso manter na memória e quais precisam ser substituídas. As páginas são substituídas quando a memória livre se torna escassa. Ademais, a maioria das páginas que contêm códigos e dados do kernel não podem ser substituídas, apenas páginas da região de endereço virtual podem. Essa troca acontece periodicamente por uma thread do kernel chamada kswapd(apelidada de

“the swap daemon”), que tem como tarefa manter o sistema de gerenciamento de memória operando de forma eficiente.(DEITEL; DEITEL; CHOFFNES, 2005)

3. Windows

O gerenciamento de memória virtual no Windows usa um esquema de gerenciamento baseado em páginas com tamanhos de quatro kilobytes e dois megabytes. Páginas de dados alocadas a um processo que não esteja na memória física são armazenadas nos arquivos de paginação em disco ou mapeadas diretamente para um arquivo regular em um sistema de arquivos local ou remoto. Uma página também pode ser marcada como de preenchimento com zeros sob demanda, o que inicializa a página com zeros antes que ela seja alocada, removendo assim o que tinha sido gravado anteriormente. Esse gerenciador do Windows usa um procedimento de dois passos para alocar a memória virtual. O primeiro reserva uma ou mais páginas de endereços virtuais no espaço de memória virtual. O segundo confirma a alocação atribuindo espaço de memória virtual. Mas, o Windows limita a quantidade de espaço de memória virtual que um processo consome, impondo uma cota à memória alocada que já foi confirmada.(SILBERSCHATZ; GALVIN; GAGNE, 2015)

Em relação ao gerenciamento da memória do SO Windows, uma aplicação chamada VirtualAlloc() e VirtualFree() são usados para habilitar a aplicação e especificar o endereço virtual em que a memória está alocada. O VirtualAlloc() é usado para reservar ou confirmar a memória virtual. O VirtualFree() serve para desconfirmar ou liberar a memória. Tais aplicações podem operar com múltiplos tamanhos de páginas de memória. Outro modo de uma aplicação usar e gerenciar memória é mapeando um arquivo para memória em seu espaço de endereçamento. O mapeamento dessa memória também é uma forma de dois processos compartilharem memória, para que os dois mapeiem o mesmo arquivo para sua memória virtual.(SILBERSCHATZ; GALVIN; GAGNE, 2015)

Em relação à fragmentação de memória do Windows, o Sistema Operacional adota principalmente duas estratégias, que são a copy-on-write e a de demanda em cluster. A paginação por demanda em cluster acontece de maneira em que o sistema de arquivos do Windows divide o espaço do disco em cluster de bytes, que são basicamente aglomerados de bytes. O Windows tira proveito da localidade espacial carregando todas as páginas no mesmo cluster de uma só vez. Esse mecanismo reduz o tempo de busca de disco porque apenas uma busca de disco é necessária para buscar o cluster inteiro.(DEITEL; DEITEL; CHOFFNES, 2005)

Já a estratégia do copy-on-write(ou também conhecida como shadowing), inicialmente mantém uma cópia de cada página compartilhada na memória. Se um processo tentar modificar a página que foi compartilhada na memória, o sistema operacional cria uma cópia da página, aplicando a modificação e atribuindo a nova cópia para esse processo no seu espaço de endereço virtual. A cópia não modificada da página permanece no espaço de endereço virtual de todos os outros processos que estão com a página compartilhada. Isso garante que, quando um processo modifica uma página compartilhada, nenhum outro processo é afetado. A copy-on-write acelera a criação do processo e reduz o consumo de memória para processos que não modificam uma quantidade significativa de dados durante a execução. No entanto, o copy-on-write pode resultar em baixo desempenho se uma parte significativa dos dados compartilhados do processo são modificados durante a

execução do programa. Neste caso, o processo sofrerá uma falha de página cada vez que modificar uma página ainda compartilhada.(DEITEL; DEITEL; CHOFFNES, 2005)

4. MacOS

Em relação à alocação de memória no Mac Os, existem diversos modos de se alocar memória virtual. Essa escolha depende tanto do motivo pelo qual a memória está sendo alocada quanto pelo lugar da chamada. Antes de relacionar os modos de alocação de memória virtual o conceito do que é o I/O é importante para o entendimento geral. O kit de I/O é uma coleção de frameworks, bibliotecas e ferramentas para criar drivers de dispositivos em OS X. Esse kit é baseado em programação orientada a objetos que se restringe a linguagem C++, que omite recursos inadequados para uso em um kernel com múltiplas threads. O kit de I/O pode apoiar recursos como multitarefas preventivas, multi-processamentos simétricos, abstrações comuns compartilhadas entre dispositivos e uma experiência de desenvolvimento aprimorada. Em geral, o kit I/O é um procedimento mais elaborado e avançado, que não é obrigatoriamente usado na alocação do Mac Os.(APPLE..., 2013)

Na linguagem C++, a biblioteca `libkern/OSMalloc.h` define algumas funções que ajudam o kernel a alocar memória, e entre elas estão a `OSMalloc()` (com as variantes `OSMalloc_noblock()` e a `OSMalloc_nowait()`), que aloca um bloco de memória, a `OSFree`, que libera a memória alocada pelo `OSMalloc()` e qualquer uma das suas variantes. Além disso, o `OSMalloc_tagalloc()` permite que seja criada uma tag para a memória alocada, e é necessário criar pelo menos uma tag antes de usar qualquer função `OSMalloc()`. Por último, a função `OSMalloc_tagfree()` libera a tag que foi alocada com o `OSMalloc_tagalloc()`.(APPLE..., 2013)

Contudo, também é possível alocar memória pelo kit I/O. Mas, as rotinas e passos que são usados no kit, em geral, não devem ser usados fora do mesmo.

O em relação a mecanismos de fragmentação, o SO MacOS usa um recurso chamado "Compressed Memory" (Memória Comprimida), que permite que o sistema operacional comprima páginas de memória antes de armazená-las no disco de paginação. Isso pode ajudar a reduzir a pressão sobre o arquivo de paginação, liberando espaço para alocar mais memória. A compressão de memória pode mitigar a fragmentação externa, pois permite que o sistema utilize de forma mais eficiente o espaço disponível.(APPLE..., 2013)

Além dele, o Mac também usa, dependendo da sua versão, a estratégia de swapping, que quando o sistema operacional deseja liberar memória física para uso por outros processos ou para evitar o esgotamento da memória, ocorre a troca do swapping. O sistema operacional pode decidir mover parte das páginas de memória de um processo para o disco ou trocar o espaço quando detecta que o processo não está usando ativamente sua memória alocada, é necessário. Devido à necessidade do sistema operacional de ler a página do disco e atualizar a tabela de páginas do processo, isso pode resultar em um atraso de falha de página.(APPLE..., 2013)

5. Discussão

Nessa análise, foi visto que dois dos sistemas operacionais, o Linux e o MacOS, podem usar a ferramenta de swapping, diferentemente do Windows, que utiliza principalmente

do copy-on-write e de demandas em cluster. O Swapping acaba por ser mais simples de ser implementado por demandar de menos passos e processos para que a paginação seja concluída. Além disso, também foi visto que o Linux e o Windows tem páginas e frames de tamanhos fixos, diferentemente do MacOS. Ademais, o MacOS tem diversas funções a mais para serem implementadas no código relacionado a alocação de memória virtual do que o Linux e o Windows.

Também foi possível perceber que o tamanho do endereço de memória virtual do Linux é a maior dentre os três sistemas operacionais, podendo suportar mais de um petabyte(o equivalente a 1000 terabytes).

6. Conclusão

Nesta prova de Infraestrutura de Software, tive muita dificuldade em achar mais artigos além dos livros já indicados pelo professor, pois sempre fui acostumado a procurar artigos e pesquisar sobre informações e outras coisas em português, o que após essa prova vai ser algo diferente, pelo resultado visto. E com essa atividade, percebi o quanto importante é a pesquisa em uma língua mais falada como o inglês, pois existe muito mais material de qualidade(e quantidade). Outra dificuldade foi como organizar o texto de maneira correta e escolher o que colocar no texto, pois grande parte do material encontrado é extenso e denso, e é impossível colocar tudo o que foi encontrado. Além de ficar muito difícil saber pelo enunciado o que colocar exatamente, pelo mesmo fato de existirem muitos materiais, tanto artigos como documentações etc. Por fim, a procura de artigos e materiais relacionados ao MacOS foi a mais complicada dos 3 sistemas operacionais pedidos. Em geral, gostei da atividade pois estimula a pesquisa e a leitura, além de ajudar também no desenvolvimento e aprendizado geral do aluno, não somente na cadeira de Infra de Software, porque é algo que será feito não só pelo resto do curso como também pelo resto da vida.

7. ChatGPT

Pergunta 1:

Me explique o conceito de memória virtual

Resposta 1:

A memória virtual é um conceito utilizado em sistemas operacionais para fornecer aos processos a ilusão de que têm acesso a uma quantidade de memória maior do que a memória física disponível no sistema. Ela permite que os processos tenham seu próprio espaço de endereçamento virtual, independente da quantidade real de memória física presente.

Na memória virtual, o espaço de endereçamento de um processo é dividido em unidades chamadas de páginas. Da mesma forma, a memória física é dividida em blocos correspondentes chamados de quadros de página. O sistema operacional realiza o mapeamento entre as páginas virtuais e os quadros de página físicos.

Quando um processo faz referência a um endereço de memória virtual, o hardware do sistema operacional traduz esse endereço para um endereço físico correspondente na memória física. Se a página necessária não estiver presente na memória física no momento, ocorre uma falta de página (page fault) e o sistema operacional busca a página

necessária no armazenamento secundário (como um disco rígido) e a traz para a memória física. O processo é então retomado e pode acessar a página solicitada.

A memória virtual oferece várias vantagens. Ela permite que os processos tenham acesso a mais memória do que a quantidade física disponível, o que pode ser essencial para a execução de programas que exigem muita memória. Além disso, o sistema operacional pode utilizar técnicas como o "swapping" (troca) de páginas entre a memória física e o armazenamento secundário para otimizar o uso da memória física, trazendo apenas as páginas necessárias para a memória quando são efetivamente utilizadas, reduzindo assim o desperdício de recursos.

No entanto, a utilização da memória virtual também pode levar a possíveis impactos no desempenho, como a ocorrência de faltas de página, que exigem operações de leitura e escrita no armazenamento secundário, o que é mais lento em comparação com o acesso à memória física. Portanto, o gerenciamento eficiente da memória virtual é fundamental para garantir um bom desempenho do sistema operacional.

Pergunta 2:

Me explique o conceito de fragmentação de memória virtual

Resposta 2:

A fragmentação da memória virtual ocorre quando o espaço disponível na memória virtual de um sistema operacional é dividido em pequenos blocos não contíguos, dificultando a alocação eficiente da memória pelos processos.

Existem dois tipos principais de fragmentação de memória virtual:

Fragmentação interna: A fragmentação interna ocorre quando um processo reserva um bloco de memória maior do que realmente precisa. Isso ocorre porque os sistemas operacionais geralmente alocam memória em unidades fixas, como páginas ou blocos de tamanho fixo. Se um processo solicitar uma quantidade menor de memória do que o tamanho da unidade alocada, o restante do espaço será desperdiçado e não poderá ser usado por outros processos. Isso resulta em fragmentação interna, onde há lacunas de espaço não utilizadas dentro dos blocos alocados.

Fragmentação Externa: A fragmentação externa ocorre quando existem lacunas de espaço livre na memória virtual, mas essas lacunas não são contíguas, ou seja, não estão próximas umas das outras. Essas lacunas podem ser resultado de processos continuamente alocando e liberando memória por processos. À medida que os processos são alocados e deslocados, a memória pode ser dividida em pequenos intervalos, impossibilitando a alocação de blocos maiores para novos processos, mesmo que haja memória livre disponível globalmente. A fragmentação externa pode levar ao uso ineficiente da memória.

A fragmentação da memória virtual pode levar a problemas de desempenho e limitações na capacidade do sistema. Pode causar atrasos na busca de espaço livre para alocar novos processos, aumentar o número de operações de troca entre a memória física e o disco rígido, diminuir a eficiência da memória cache e dificultar o gerenciamento e uso adequado da memória disponível.

Para mitigar a fragmentação da memória virtual, os sistemas operacionais em-

pregam técnicas como compactação (reorganização dos blocos de memória para eliminar lacunas) e algoritmos eficientes de alocação de memória, como Best Fit (alocação no menor bloco que atende aos requisitos), Next Fit (alocação no próximo bloco após o último alocado) ou First Fit (alocação no primeiro bloco disponível que atenda aos requisitos). Essas estratégias visam reduzir a fragmentação e otimizar o uso da memória virtual.

Pergunta 3:

Como funciona a alocação do espaço de memória no Sistema Operacional macOS?

Resposta 3:

A alocação do espaço de memória no sistema operacional macOS é gerenciada pelo kernel do macOS, que implementa um modelo de memória virtual. O macOS utiliza uma técnica chamada "paging" (paginamento) para dividir a memória em páginas de tamanho fixo. O tamanho padrão da página no macOS é de 4 KB.

Quando um processo é iniciado no macOS, ele recebe um espaço de endereçamento virtual, que é dividido em páginas virtuais. O sistema operacional aloca memória física e mapeia essas páginas virtuais para a memória física, em um processo conhecido como mapeamento de páginas. No entanto, a alocação real da memória física ocorre sob demanda, ou seja, as páginas virtuais são mapeadas para a memória física apenas quando são realmente acessadas pelo processo.

O macOS utiliza um esquema de memória virtual chamado "demand paging" (paginamento sob demanda). Isso significa que, quando um processo tenta acessar uma página virtual que ainda não foi mapeada para a memória física, ocorre uma falha de página. O sistema operacional então busca a página necessária no disco rígido, no arquivo de paginação (swap file), e a traz para a memória física. Esse processo é conhecido como operação de "page fault" (falha de página).

Além disso, o macOS oferece suporte a recursos avançados de gerenciamento de memória, como compressão de memória (Memory Compression) e compactação de memória (Memory Compact). Essas técnicas visam otimizar o uso da memória e melhorar o desempenho do sistema, especialmente em situações de escassez de memória.

No geral, a alocação do espaço de memória no macOS envolve o mapeamento de páginas virtuais para a memória física sob demanda, usando um sistema de paginação e técnicas avançadas de gerenciamento de memória para otimizar o uso da memória e garantir um bom desempenho do sistema.

8. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in parenthesis, e.g. (SILBERSCHATZ; GALVIN; GAGNE, 2015), (DEITEL; DEITEL; CHOFFNES, 2005), and (APPLE..., 2013).

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

References

APPLE Documentation. 2013. 31/05. Disponível em: https://developer.apple.com/library/archive/documentation/Darwin/Conceptual/KernelProgramming/vm/vm.html#/apple_ref/doc/uid/TP30000905-CH210-CHDHIIJF.

DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. **Sistemas operacionais**. 3th. ed. [S.l.]: Pearson Education do Brasil, 2005.

MICROSOFT Documentation. 2021. 31/05. Disponível em: <https://learn.microsoft.com/en-us/windows/win32/memory/working-set>.

MICROSOFT Documentation. 2021. 31/05. Disponível em: <https://learn.microsoft.com/pt-br/cpp/c-runtime-library/reference/malloc?view=msvc-170>.

SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Fundamentos de Sistemas operacionais**. 9th. ed. [S.l.]: LTC — Livros Técnicos e Científicos Editora Ltda., 2015.