

# Substituição de Páginas

Pedro H. Penna, Henrique Freitas,  
Márcio Castro and Jean-François Méhaut

Pontifical Catholic University of Minas Gerais  
Federal University of Santa Catarina  
University of Grenoble Alpes

## Resumo

A memória de um computador é organizada em uma hierarquia, com o sistema operacional atuando como gerente. Nessa organização, uma função fundamental é a de manter páginas utilizadas em memória e descarregar as não necessárias em disco. No Nanvix, essa tarefa é de responsabilidade do substituidor de páginas e nesse projeto você deverá propor melhorias a ele.

## Fundamentação Teórica

Para proporcionar a ilusão de um computador com memória infinitamente grande, rápida e não volátil, a memória de um computador é organizada em uma hierarquia, com memórias pequenas, voláteis e de acesso rápido nos primeiros níveis, e memórias lentas, densas e não voláteis nos últimos.

Nessa hierarquia, o papel do sistema operacional é o de realizar a gerência de maneira eficiente e transparente ao programador. Então, para alcançar esse objetivo, sistemas operacionais modernos, recorrem a uma técnica híbrida de *hardware* e *software*, denominada memória virtual.

Nessa técnica, o espaço de endereçamento de um processo é dividido em regiões de mesmo tamanho, denominadas páginas, que, por sua vez, são mapeados em regiões de tamanho correspondente na memória física, denominadas quadros de página. Assim, um conjunto de processos pode compartilhar a memória física, mesmo que não caibam integralmente nela.

Para tanto, tudo o que o sistema operacional deve fazer é manter as páginas mais utilizadas na memória, armazenando em disco aquelas que não são mais necessárias. Essa gerência exerce influência direta no desempenho final do sistema e é de responsabilidade do substituidor de páginas, um componente do módulo de gerenciamento de memória.

## Descrição do Projeto

O componente de substituição de páginas do Nanvix adota uma política local *first-in-first-out*. Isto é, quando uma falta de página ocorre, o sistema operacional escolhe, dentro das páginas residentes em memória do processo que falhou, a página mais antiga para ser substituída. Essa política tem como principais vantagens a simplicidade na implementação e *overhead* mínimo na operação de substituição em si.

No entanto, essa política impõe sérios problemas de desempenho ao sistema, principalmente se processos possuírem um padrão de acesso aos dados – o que geralmente é verdade. Por exemplo, suponha um programa que efetue a multiplicação de duas matrizes grandes, que não cabem na memória. Nesse caso, manter em memória as páginas novas certamente não é a escolha mais sábia, mas sim manter as páginas que representem o conjunto de trabalho do processo, que são as linhas e colunas das matrizes operando e partes da matriz resultado.

Para resolver esse problema, você deverá modificar o algoritmo de substituição de páginas do Nanvix e implementar qualquer outra política que conduza a melhores desempenhos. Você deverá avaliar o desempenho da sua solução quantitativamente. Para isso, você pode utilizar um código exemplo que efetua a multiplicação de matrizes.

**ATENÇÃO:** A nota máxima a ser atribuída na implementação dependerá do algoritmo de substituição de páginas implementado: algoritmo do relógio (7,0), NRU (8,0), NFU (9,0), Aging ou WSClock (10,0).

## Por Onde Começar?

O código do subsistema de memória do Nanvix está no diretório `kernel/mm`, dividido em vários arquivos:

- `mm.c`: inicialização do gerenciador de memória.
- `paging.c`: sistema de paginação e *swapping*.
- `region.c`: gerenciamento de regiões.

Nesse projeto, você deve se atentar ao arquivo `paging.c`, principalmente na função `allocf()`, que implementa a função de substituição de páginas. Além disso, desses arquivos, você também usar o exemplo de multiplicação de matrizes em `src/sbin/test/test.c`.

## Testes Básicos

O Nanvix possui um programa que permite realizar alguns testes de escalonamento, entrada/saída, comunicação entre processos e paginação denominado `test`. O seu código fonte está localizado no diretório `src/sbin/test`.

Para realizar os testes de paginação execute `test swp` a partir do terminal do Nanvix. A execução deverá demorar um certo tempo. Esse programa utilitário realizará alguns testes de stress envolvendo o sistema de gerenciamento de memória do kernel.

Por padrão, o `test swp` realiza um teste mais completo, podendo levar vários minutos para terminar. Para fazer um teste menos completo, e portanto, mais rápido, modifique a linha `static unsigned flags = VERBOSE | FULL` para `static unsigned flags = VERBOSE | EXTENDED`. Caso você sinta necessidade de exercitar outras funcionalidades do seu algoritmo, fique à vontade para modificar esse utilitário.

## Grupos, Avaliação e Entrega

O trabalho deverá ser realizado necessariamente em grupos de 3 alunos. Apenas um dos integrantes de cada grupo deverá enviar através do Moodle um arquivo em formato ZIP contendo:

- Código fonte do Nanvix modificado (diretório principal completo do projeto contendo as modificações feitas). **Não se esqueça de executar o comando `sudo make clean`, bem como verificar se todos os arquivos do Nanvix estão presentes, antes de compactar os arquivos.**
- Um PDF contendo uma pequena descrição da solução implementada.

Os trabalhos serão apresentados nos dias definidos no cronograma, disponível no Moodle. O professor irá avaliar tanto a corretude, desempenho e clareza da solução, quanto a apresentação feita pelo grupo. A data/hora limite para o envio dos trabalhos é **08/06/2017 às 23h55min**. Não será permitida a entrega de trabalhos fora desse prazo.