UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
SISTEMAS OPERACIONAIS

Simulador de algoritmos de substituição de páginas na memória

Bruno Dias (108349)
Davi Ribeiro (88931)
Renan Barreto (85486)

Prof. Dr. Pedro de Botelho Marcos

Rio Grande - RS
2021

1 – Introdução

O objetivo do trabalho é a implementação de diferentes algoritmos de substituição de página: FIFO, Menos Recentemente Usada (MRU), Não usada frequentemente (NUF) e ótimo, para serem empregados para escolher qual página deve ser substituída da memória. A partir de um arquivo de texto contendo múltiplas linhas, sendo uma linha referente a cada caso de teste, nos foi retornado a seguinte formatação:

Número de molduras de página na memória| Número de páginas do processo| Sequência em que as páginas são acessadas

Após inseridas entradas nos algoritmos o objetivo dos algoritmos é retornar o número de trocas necessárias para cada processo e qual algoritmo se assemelha mais ao algoritmo ótimo.

2 – Desempenho dos algoritmos

Como foi solicitado, fizemos alguns testes com o código, de acordo as entradas que nos foi cedida, que nos possibilitou responder alguns possíveis questionamentos passados.

2.1 - Como a quantidade de molduras afeta o desempenho dos algoritmos?

De forma geral, para todos os algoritmos quanto mais molduras tivermos disponíveis, mais desempenho terá nosso algoritmo, pois necessitaremos de menos trocas de páginas de acordo elas são referenciadas e mais páginas presentes quando solicitadas seu acesso. Porém especificamente no FIFO a situação é o inverso disso.

Posição da mem.	0	1	2	3	4	5	6	7	8	9	10	11	12
Moldura 1		0	0	0	3	3	3	4	4	4	4	4	**4**
Moldura 2			1	1	1	0	0	0	0	0	2	2	**2**
Moldura 3				2	2	2	1	1	1	1	1	3	**3**

Tabela 1. Solicitação de páginas FIFO 3 molduras

Posição da mem.	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Moldura 1		0	0	0	0	0	0	4	4	4	4	3	**3**
Moldura 2			1	1	1	1	1	1	0	0	0	0	**4**
Moldura 3				2	2	2	2	2	2	1	1	1	**1**
Moldura 4					3	3	3	3	3	3	2	2	**2**

Tabela 2. Solicitação de páginas FIFO 4 molduras

Entrada: 0 1 2 3 0 1 4 0 1 2 3 4

De acordo a tabela 1 podemos observar que 12 páginas foram solicitadas, mas dessas 12, 9 foram substituídas (perdidas), o que dá uma taxa de 75% de falta de páginas e somente 25% de sucesso ao se encontrar essas páginas. Na tabela 2 podemos ver que a falta de páginas sobe para 10 das mesmas 12, aumentando assim as páginas substituídas e perdidas, diminuindo ainda mais a chance de sucesso ao encontrar uma página. Isso chama-se anomalia de Belady, e demonstra que para o FIFO em especial quantos mais molduras mais páginas perdidas teremos, e isso afeta diretamente no desempenho que o algoritmo pode apresentar em um ambiente onde se há um grande número de páginas comparados a molduras.

2.2 - Como a quantidade de páginas afeta o desempenho dos algoritmos?

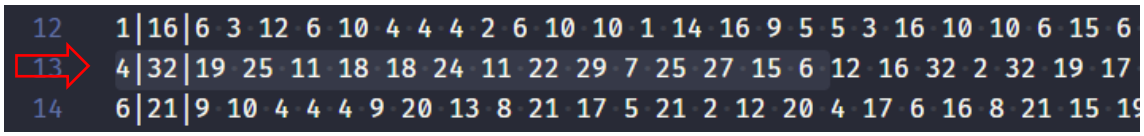

Figura 1. Arquivo de entrada

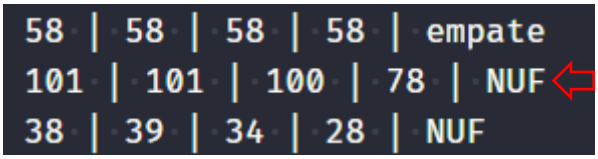

Figura 2. Saída dos algoritmos

De acordo com as entradas da figura 1, na linha 13 temos 32 páginas que são referenciadas 112 vezes, já na linha 14 temos 21 páginas e bem menos da metade de páginas referenciadas. É possível constatar na saída dos algoritmos na figura 2, que quanto mais páginas tivermos, mais longe do ótimo os algoritmos se encontrarão. Portanto quanto mais páginas, mais trocas e menos otimizado se tornam os algoritmos testados.

2.3 - Existe alguma relação entre o número de molduras e o número de páginas que afeta o desempenho dos algoritmos?

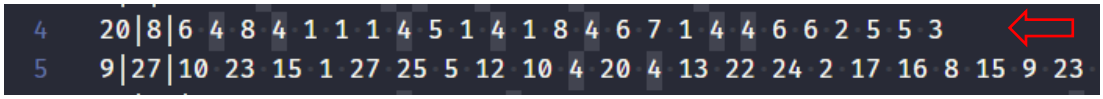

Figura 3. Arquivo de entrada

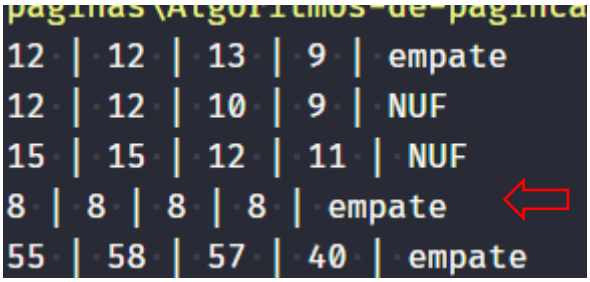

Figura 4. Saída dos algoritmos

Como podemos reparar na figura 3 linha 4, temos um número de molduras bem maior que o número de páginas e na figura 4 na mesma linha, onde obtivemos as saídas dos algoritmos, podemos verificar que houve um empate, pois todos os algoritmos resultaram no desempenho do algoritmo ótimo. Então podemos chegar à conclusão que há uma correlação de desempenho quanto ao número de molduras e páginas, como foi observado.

2.4 - Como a ordem de referenciamento das páginas afeta o desempenho dos algoritmos?

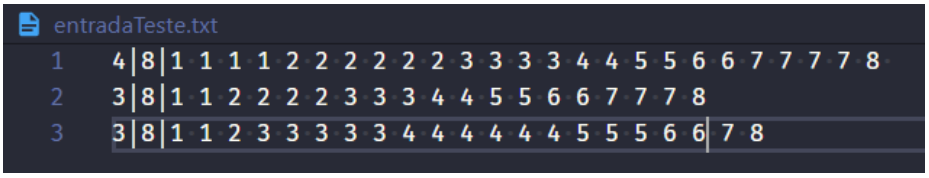

Figura 5. Entrada ordenada

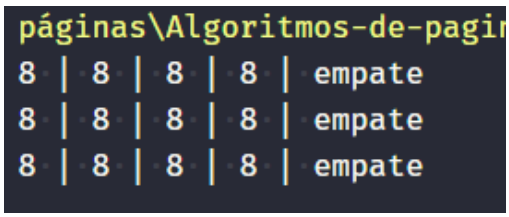

Figura 6. Saída algoritmos entrada ordenada

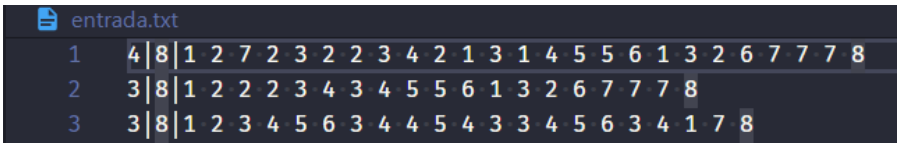

Figura 7. Entrada desordenada

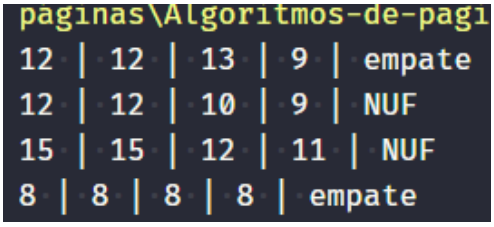

Figura 8. Saída algoritmos entrada desordenada

Para responder esse questionamento podemos observar a sequência de figuras acima. Na figura 5 ordenamos as entradas das três primeiras linhas do arquivo que nos foi cedido pelo professor e na figura 6, obtivemos o resultado das saídas dessas entradas ordenadas. Na figura 7 temos a entrada original e na figura 8 a saída com entrada original. Para as entradas ordenadas obtivemos todos os resultados igual ao algoritmo ótimo. Portanto, em um cenário ideal onde tudo fosse ordenado, todos nossos algoritmos teriam desempenho ótimo, pois a ordem de referenciamento influencia nesse caso em termos menos trocas, uma sequência seguida de consultas às páginas e sem problema de consultar páginas que possivelmente fossem substituídas ou perdidas.