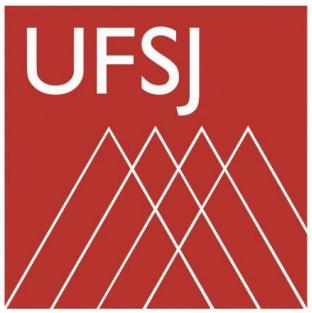


Universidade Federal de São João del-Rei

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI

Segundo Trabalho Prático da Disciplina de Sistemas Operacionais

Documentação referente ao segundo trabalho prático da disciplina Sistemas Operacionais 2019/2, desenvolvido pelos alunos Júlio Hebert Silva e Tomaz Miranda de Oliveira do curso de Ciência da Computação da UFSJ.

São João Del Rei
Novembro/2019

1 Introdução

Este trabalho tem como objetivo descrever a implementação de alguns algoritmos de substituição de páginas, que foi realizada na linguagem C.

2 Entrada

A entrada do programa acontece por meio de um arquivo que é passado por parâmetro ao executar o programa, assim como outras informações importantes tais como, qual algoritmo utilizar, tamanho da página e tamanho da memória. Caso arquivo passado exista, será lida linha por linha até o fim, e cada linha deve conter um endereço físico em hexadecimal e um comando especifico, como ‘R’ e ‘W’. O algoritmo, caso a linha lida não possua comando ele irá utilizar o mesmo do último comando passado

2.1 Formado da Entrada

Após formatação do texto recebido é hora de realizar o processamento do vetor para identificar o comando especial contido. Tais comando são:

- W – Esse indica que a página referente aquele endereço será alterada pelo processo que a requisitou.
- R - Indica que essa página será lida pelo processo.

3 Execução

Assim que o programa é executado, é passado para ele o nome do algoritmo a ser utilizado, o nome do arquivo de entradas, tamanho de páginas e tamanho da memória.

Tendo isso em mente, os três algoritmos são primeiramente implementador com uma Hash bem definida com as informações da página, tais como addr(já shifitado), um bit R, bit M e o tempo de último acesso. Esse tempo começa em zero e é implementado a cada leitura de linha do arquivo.

Caso o algoritmo a ser utilizado seja o LRU, o programa então cria uma lista duplamente encadeada com apenas o endereço da página, para termos o controle das páginas referenciadas. Quando uma página é requisitada essa lista insere os endereços na primeira posição, empurrado as demais posições para traz, mas antes disso, verifica começando pela primeira posição - já que uma página referenciada tem mais chance de estar entre as últimas referenciada - procurando pelo endereço, caso ele esteja na lista o removemos dela e é adicionado no começo novamente. Em paralelo o Hash naquela posição estão sendo feitas as devidas mudanças em seus bit R/M e no tempo de último acesso.

Com isso, caso haja uma falta de página o algoritmo de LRU apenas remove a última posição da lista, tendo em vista que é a página a mais tempo sem ter sido referenciada, fazendo com que o algoritmo em si trabalhe num tempo constante.

Seguindo essa lógica, caso algoritmo selecionado seja NRU, as primeiras páginas serão adicionadas normalmente na Hash e caso haja a falta de página, o algoritmo NRU vai percorrer toda a Hash atrás da página de menor Classe ou até encontrar uma página de Classe 1. Note que de tempos em tempos o programa sempre altera o bit R de páginas mais velhas para 0.

E caso o algoritmo seja segunda_chance, teremos uma lista circular para armazenar os addr das páginas para controlarmos em qual posição estamos para selecionar a página propicia a ser removida. Sempre que uma nova página é referenciada, além dela ser armazenada no Hash, seu endereço é salvo na lista, para caso haja uma falta de página, o algoritmo sempre saberá a próxima página que possivelmente será substituída.

Ao final da execução, teremos um leve resumo com informações tais como, número de páginas lidas da memória, páginas que foram escritas nos disco novamente e o tempo de execução em ms.

4 Analises

1 Análise de desempenho:

1.1 - Utilizando o arquivo compilador.log

- Utilizando técnica de reposição LRU

Tempo gasto:

Mem/Pág	2	4	8	16	32	64
256	1145 ms	1091 ms	1051 ms	1012 ms	997 ms	988 ms
1024	1486 ms	1235 ms	1129 ms	1042 ms	1000 ms	984 ms
4096	4881 ms	2098 ms	1458 ms	1158 ms	1040 ms	999 ms
16384	17376 ms	10544 ms	4748 ms	2017 ms	1261 ms	1098 ms

- Utilizando tecnica de reposição NRU

Tempo gasto:

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
256	1809 ms	2074 ms	2094 ms	2109 ms	2133 ms	2204 ms
1024	2475 ms	1961 ms	1799 ms	1556 ms	1592 ms	1654 ms
4096	6237 ms	4183 ms	4729 ms	1623 ms	1316 ms	1223 ms
16834	32929 ms	14144 ms	6086 ms	2748 ms	1675 ms	1360 ms

- Utilizando tecnica de reposição Segunda Chance

Tempo gasto:

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2	958 ms	939 ms	939 ms	899 ms	880 ms	887 ms
2	979 ms	959 ms	925 ms	869 ms	842 ms	841 ms
2	1117 ms	1022 ms	938 ms	873 ms	828 ms	806 ms
2	1893 ms	1349 ms	989 ms	892 ms	843 ms	806 ms

1.2 – Usando o arquivo compressor.log

- Utilizando tecnica de reposição LRU

Tempo gasto:

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2^8	1172 ms	1038 ms	1011 ms	984 ms	946 ms	933 ms
2^{10}	1261 ms	1183 ms	1155 ms	1018 ms	940 ms	926 ms
2^{12}	1504 ms	1358 ms	1251 ms	1040 ms	962 ms	936 ms
2^{14}	2206 ms	1813 ms	1463 ms	1108 ms	1000 ms	942 ms

- Utilizando tecnica de reposição NRU

Tempo gasto:

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2^8	1034 ms	1155 ms	1665 ms	1473 ms	1371 ms	1567 ms
2^{10}	1211 ms	1123 ms	1045 ms	919 ms	901 ms	993 ms
2^{12}	2325 ms	1591 ms	1207 ms	970 ms	863 ms	805 ms
2^{14}	2899 ms	6377 ms	2358 ms	1322 ms	956 ms	821 ms

- Utilizando a tecnica de reposição Segunda Chance

Tempo gasto:

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2^8	747 ms	838 ms	847 ms	803 ms	806 ms	792 ms
2^{10}	777 ms	754 ms	739 ms	722 ms	762 ms	749 ms
2^{12}	835 ms	811 ms	760 ms	726 ms	713 ms	706 ms
2^{14}	946 ms	843 ms	791 ms	749 ms	732 ms	723 ms

1.3 – Utilizando o arquivo matriz.log

- Utilizando a tecnica de reposição LRU

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2^8	1266 ms	1200 ms	1056 ms	1017 ms	994 ms	977 ms
2^{10}	1445 ms	1333 ms	1242 ms	1096 ms	1018 ms	983 ms
2^{12}	2494 ms	1733 ms	1415 ms	1199 ms	1072 ms	1011 ms
2^{14}	7533 ms	4964 ms	2347 ms	1510 ms	1181 ms	1055 ms

- Utilizando a tecnica de reposição NRU

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2^8	1549 ms	1755 ms	2320 ms	2258 ms	2263 ms	2313 ms
2^{10}	2430 ms	1758 ms	1509 ms	1261 ms	1445 ms	1769 ms
2^{12}	6815 ms	3605 ms	2244 ms	1444 ms	1183 ms	1010 ms
2^{14}	29584 ms	14167 ms	6207 ms	2671 ms	1575 ms	1164 ms

- Utilizando a tecnica de reposição Segunda Chance:

Tempo gasto:

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2^8	886 ms	916 ms	969 ms	916 ms	887 ms	888 ms
2^{10}	908 ms	891 ms	851 ms	811 ms	830 ms	841 ms
2^{12}	1126 ms	971 ms	868 ms	815 ms	777 ms	752 ms
2^{14}	2176 ms	1444 ms	977 ms	859 ms	790 ms	751 ms

1.4 – Utilizando o arquivo simulador.log

- Utilizando a tecnica de reposição LRU

Tempo gasto:

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2^8	1189 ms	1124 ms	1058 ms	1019 ms	1002 ms	991 ms
2^{10}	1579 ms	1317 ms	1166 ms	1063 ms	1001 ms	977 ms
2^{12}	3380 ms	2088 ms	1551 ms	1237 ms	1053 ms	1000 ms
2^{14}	8232 ms	5325 ms	3237 ms	1805 ms	1246 ms	1091 ms

- Utilizando a tecnica de reposição NRU

Tempo gasto::

Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2^8	1711 ms	1940 ms	2301 ms	2018 ms	1906 ms	1957 ms
2^{10}	2584 ms	1912 ms	1722 ms	1447 ms	1497 ms	1626 ms
2^{12}	6731 ms	3761 ms	2528 ms	1604 ms	1295 ms	1127 ms
2^{14}	29183 ms	13702 ms	6403 ms	2914 ms	1756 ms	1275 ms

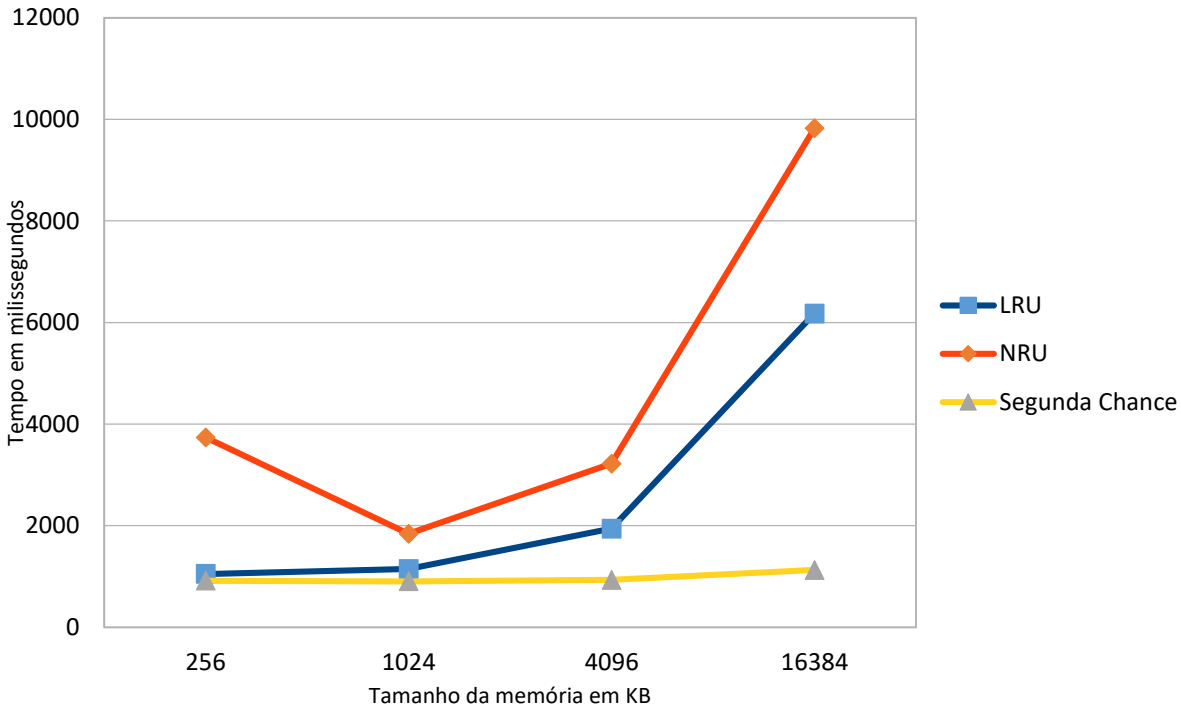
- Utilizando a tecnica de reposição Segunda Chance

Tempo gasto:

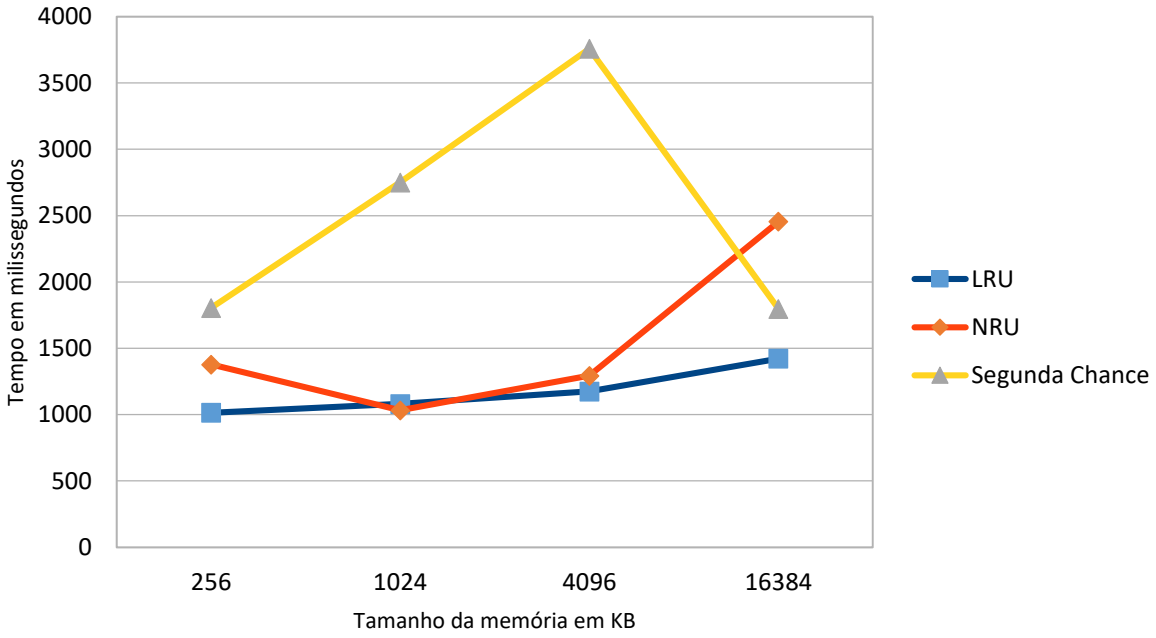
Mem/Pág	2^1	2^2	2^3	2^4	2^5	2^6
2^8	906 ms	916 ms	953 ms	908 ms	883 ms	898 ms
2^{10}	909 ms	907 ms	882 ms	841 ms	820 ms	824 ms
2^{12}	1081 ms	955 ms	875 ms	824 ms	799 ms	776 ms
2^{14}	1877 ms	1354 ms	958 ms	854 ms	797 ms	767 ms

Grafico de comparação entre os métodos de acordo com o arquivo dado:

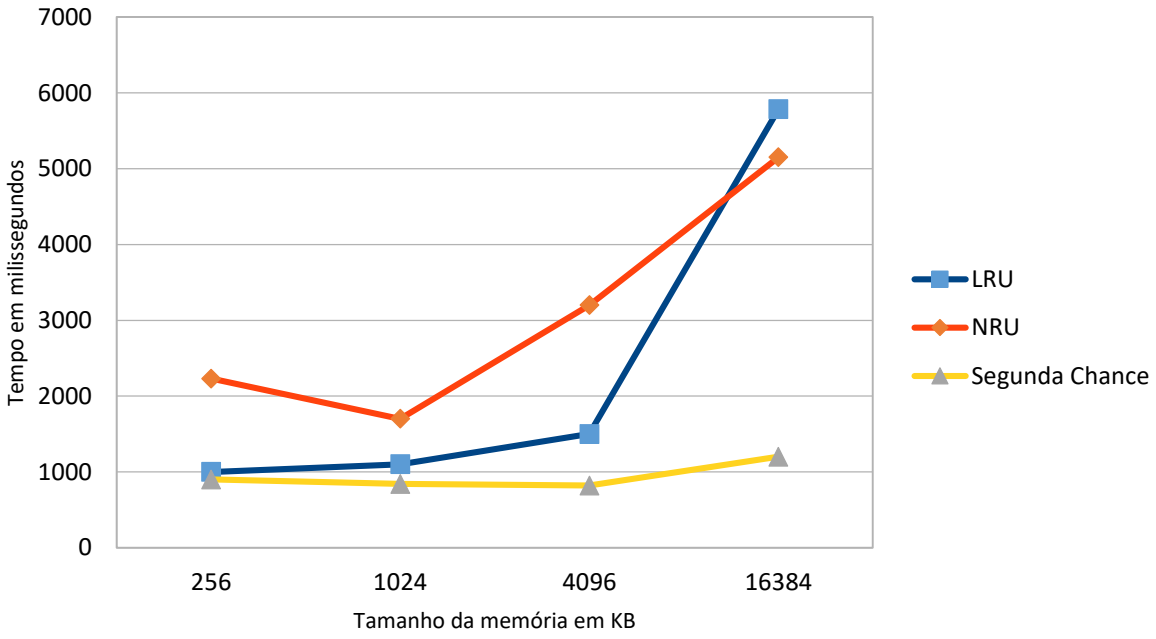
❿ arquivo compilador.log:



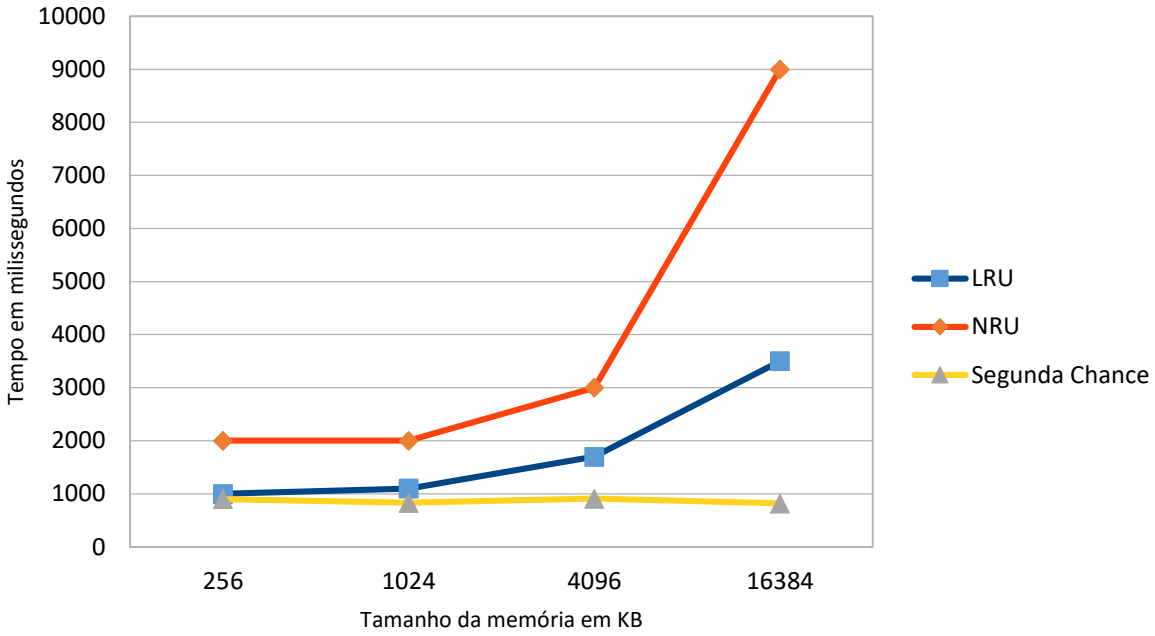
❿ arquivo compressor.log:



❿ arquivo matriz.log



❿ arquivo simulador.log:



5 Considerações Finais

Neste trabalho prático, através da implementação simplificada de alguns algoritmos de substituição de páginas, pudemos aprender um pouco mais como funciona na prática no sistema operacional. Além de rever conceitos importantes para computação como a lista e a hash.