

# Relatório sobre algoritmos de substituição de páginas

Lucas Daher Santos (RA:114830) e Daniel Barbosa Silva Costa (RA:112185)

Junho 2019

Para a realização dos testes, foram escolhidos dois tipos de algoritmos de substituição de páginas:

- *Least Recently Used (LRU)*
- *Not Recently Used (NRU)*

Os algoritmos foram implementados em linguagem Python (Versão 3.7). Para cada algoritmo, foram testadas 4 configurações diferentes (tamanhos de memórias e página), cada uma testada 4 vezes a fim de retirar uma média para cada configuração. Os resultados e algoritmos utilizados estão disponibilizados no Apêndice.

Na *main*, basta substituir os valores para aqueles desejados e selecionar o algoritmo de substituição desejado "*LRU*" ou "*NRU*" em *ram.allocate()* na linha 324 da Apêndice A1. Para observar de fato o funcionamento, é possível alterar o valor da variável *debug* para 1 (ativar o *debug* ocasionará lentidão devido aos *outputs*).

Os arquivos .txt foram gerados através de escritas em arquivos. Os 4 testes, contendo 4 execuções cada, para o mesmo algoritmo com as mesmas configurações foram salvos em um arquivo, em 8 logs de .txt (4 para cada um dos dois algoritmos, cada um desses com sua configuração). A partir disso, foi utilizado o algoritmo de calculo das médias, indicando na variável *t* o arquivo a ser gerado as médias (o algoritmo foi feito exatamente para as saídas anteriormente obtidas e o resultado anexado no próprio arquivo).

O tempo de execução foi obtido através do armazenamento do tempo decorrido até o início de fato do algoritmo de substituição e do momento exato da finalização do algoritmo, então fez-se a subtração para obtenção do tempo gasto executando apenas o algoritmo de substituição. Os tempos foram gerenciados em nanosegundos e convertidos para segundos

## *Least Recently Used (LRU)*

O algoritmo de substituição *Least Recently Used (LRU)* trata a memória RAM de forma similar a uma fila, sempre que uma página buscar pela memória RAM, essa será devidamente alocada no final de uma fila, a qual demarca utilizações (o início possui o menos recentemente usado recentemente e o final o mais recente mente), e o primeiro elemento será retirado (o menos usado recentemente). Porém, caso uma página, no momento em que solicita presença na RAM, já esteja referenciada em uma das molduras, não há novas inserções ou remoções da fila, apenas o seu valor na fila será transferido para o final, como se tivesse acontecido uma inserção.

A seguir são apresentadas as médias dos testes, bem como suas configurações

### Teste 1

Configurações:

- Page size: 4.0
- Physical memory: 128.0
- Physical memory page frames: 32.0
- Virtual memory: 1024.0
- Logical memory virtual pages: 256.0

Resultados (Média):

- Page faults: 239.0
- Referenced (Replaced): 207.0
- Unreferenced (Replaced): 0.0
- Modified (Replaced): 52.75
- Unmodified (Replaced): 154.25
- Time: 0.001926 seconds

### Teste 2

Configurações:

- Page size: 8.0
- Physical memory: 128.0
- Physical memory page frames: 16.0
- Virtual memory: 1024.0

- Logical memory virtual pages: 128.0

Resultados (Média):

- Page faults: 119.5
- Referenced (Replaced): 103.5
- Unreferenced (Replaced): 0.0
- Modified (Replaced): 25.5
- Unmodified (Replaced): 78.0
- Time: 0.00069325 seconds

### Teste 3

Configurações:

- Page size: 32.0
- Physical memory: 128000.0
- Physical memory page frames: 4000.0
- Virtual memory: 1024000.0
- Logical memory virtual pages: 32000.0

Resultados (Média):

- Page faults: 29164.5
- Referenced (Replaced): 25164.5
- Unreferenced (Replaced): 0.0
- Modified (Replaced): 6246.5
- Unmodified (Replaced): 18918.0
- Time: 15.892020925 seconds

### Teste 4

Configurações:

- Page size: 64.0
- Physical memory: 128000.0
- Physical memory page frames: 2000.0
- Virtual memory: 1024000.0

- Logical memory virtual pages: 16000.0

Resultados (Média):

- Page faults: 14576.25
- Referenced (Replaced): 12576.25
- Unreferenced (Replaced): 0.0
- Modified (Replaced): 3138.5
- Unmodified (Replaced): 9437.75
- Time: 3.944210425 seconds

## *Not Recently Used (NRU)*

O algoritmo de substituição *Not Recently Used (NRU)* trata a memória RAM via classes de páginas:

- 0: não referenciadas e não modificadas
- 1: não referenciadas e modificadas
- 2: referenciadas e não modificadas
- 3: referenciadas e modificadas

A cada ciclo de clock, ou ciclos, há o rebaixamento de uma determinada classe. Classe 3 é rebaixada para classe 1, classe 2 é rebaixada para classe 0 e classe 1 para classe 0.

No algoritmo feito, a classe 3 demora por  $(2 \cdot \text{moduras\_de\_páginas})$  clocks para ser rebaixada, a classe 2 por  $(1.5 \cdot \text{moduras\_de\_páginas})$  clocks para ser rebaixada e a classe 1 por  $(1 \cdot \text{moduras\_de\_páginas})$  clocks para ser rebaixada. O ciclo de clock é simulado como iterações, ou seja, a cada uma iteração há um ciclo de clock simulado.

## Teste 1

Configurações:

- Page size: 4.0
- Physical memory: 128.0
- Physical memory page frames: 32.0
- Virtual memory: 1024.0
- Logical memory virtual pages: 256.0

Resultados (Média):

- Page faults: 233.75
- Referenced (Replaced): 100.25
- Unreferenced (Replaced): 101.5
- Modified (Replaced): 40.5
- Unmodified (Replaced): 161.25
- Time: 0.0040084000000000005 seconds

## Teste 2

Configurações:

- Page size: 8.0
- Physical memory: 128.0
- Physical memory page frames: 16.0
- Virtual memory: 1024.0
- Logical memory virtual pages: 128.0

Resultados (Média):

- Page faults: 119.0
- Referenced (Replaced): 52.75
- Unreferenced (Replaced): 50.25
- Modified (Replaced): 19.75
- Unmodified (Replaced): 83.25
- Time: 0.0011928 seconds

## Teste 3

Configurações:

- Page size: 32.0
- Physical memory: 128000.0
- Physical memory page frames: 4000.0
- Virtual memory: 1024000.0
- Logical memory virtual pages: 32000.0

Resultados (Média):

- Page faults: 29137.0
- Referenced (Replaced): 11458.25
- Unreferenced (Replaced): 13678.75
- Modified (Replaced): 4993.25
- Unmodified (Replaced): 20143.75
- Time: 62.908054225 seconds

## Teste 4

Configurações:

- Page size: 64.0
- Physical memory: 128000.0
- Physical memory page frames: 2000.0
- Virtual memory: 1024000.0
- Logical memory virtual pages: 16000.0

Resultados (Média):

- Page faults: 14579.5
- Referenced (Replaced): 5750.25
- Unreferenced (Replaced): 6829.25
- Modified (Replaced): 2509.0
- Unmodified (Replaced): 10070.5
- Time: 15.7670848 seconds

Partindo do princípio que todas páginas da memória física alocadas em memória física são modificadas ou não, são referenciadas ou não, é possível verificar a consistência dos dados realizando a soma de modificadas e não modificadas substituídas e igualar com a soma de referenciadas e não referenciadas, ou seja, ambas somas devem ser iguais. Além disso, essa soma acrescida do valor da quantidade de molduras na memória física deve ser igual à quantidade de falta de páginas, já que primeiramente a memória esta vazia, ou seja, sem substituição, mas há a contabilização de falta de páginas, já que a página não encontra-se na memória física.

## Conclusões

Como é possível observar, o algoritmo *LRU* demonstrou-se mais eficiente que o *NRU*:

- Teste 1: *LRU* finaliza em aproximadamente 48.15% do tempo gasto por *NRU*
- Teste 2: *LRU* finaliza em aproximadamente 58.11% do tempo gasto por *NRU*
- Teste 3: *LRU* finaliza em aproximadamente 25.20% do tempo gasto por *NRU*
- Teste 4: *LRU* finaliza em aproximadamente 25.00% do tempo gasto por *NRU*

Além disso, percebe-se uma maior velocidade com maiores tamanhos de páginas, isso pelo fato de não se obter o número de processos previamente, sendo esse calculado durante a execução com base nos tamanhos das memórias e tamanho de página. Frente a um aumento muito maior das memórias com relação ao tamanho de página, maior custo de processamento foi obtido para ambos os algoritmos, sendo o *LRU* mais eficiente.

# Apêndice

## Apêndice A - Algoritmos em Python

### Apêndice A1 - Algoritmos da simulação

```
1 import numpy as np
2 import random as rd
3 import collections
4 import time
5 import sys
6 import gc
7
8 class page():
9     def __init__(self, action, address, status, ramPosition): #action R or W
10         self.action=action
11         self.address=address
12         self.status=status
13         self.ramPosition=ramPosition
14
15 class vMemory():
16     def __init__(self, size, pageSize):
17         self.size=size
18         self.pageSize=pageSize
19         self.capacity=int(self.size/self.pageSize)
20         self.pages=[]
21
22     def generate(self): #
23         k=self.capacity
24         qtd1=list(range(1,k+1))
25         rd.shuffle(qtd1)
26         qtd2=list(range(1,k+1))
27         rd.shuffle(qtd2)
28         qtd3=list(range(1,k+1))
29         rd.shuffle(qtd3)
30         qtd4=list(range(1,k+1))
31         rd.shuffle(qtd4)
32         qtd=[*qtd1[:int(k/4)],*qtd2[int(k/4):int(2*k/4)],
33              *qtd3[int(2*k/4):int(3*k/4)],*qtd4[int(3*k/4):]]
34         rd.shuffle(qtd)
35         time.sleep(1)
36
37         #check for possible callbacks
38         '''gh=qtd.copy()
39         gh.sort()
40         print(len(qtd))
41         gk=sorted(set([x for x in qtd if qtd.count(x) > 1]))
42         gk.sort()
43         print(gk)
44         print(gh)'''
45
46         k=""
47         for i in range(self.capacity):
48             if(rd.randint(1,4)%4==0):
49                 k='W'
50             else:
51                 k='R'
52             self.pages.append(page(k, qtd[i], -1, -1))
```



```

53     print("Virtual Memory Configured")
54
55 class RAM():
56     def __init__(self, size, vMemory, pageSize, debug):
57         self.size=size
58         self.vMemory=vMemory
59         self.pageSize=pageSize
60         self.debug=debug
61         self.capacity=int(self.size/self.pageSize)
62         self.used=0
63         self.frames=[]
64         self.rw=[]
65         self.clock=[]
66         self.pageFault=0
67         self.nref=0
68         self.nmod=0
69         self.ref=0
70         self.mod=0
71
72     def setUpRAM(self):
73         print("\nConfiguring RAM")
74         for i in range(self.capacity):
75             self.frames.append(-1)
76             self.rw.append(-1)
77             self.clock.append(-1)
78         self.used=0;
79         self.pageFault=0
80         self.nref=0
81         self.nmod=0
82         self.ref=0
83         self.mod=0
84
85     def verifyOcurrency(self, address):
86         i=0
87         while(i<self.capacity):
88             if(address == self.frames[i]):
89                 break
90             else:
91                 i+=1
92         if(i<self.capacity):
93             return i
94         else:
95             return -1
96
97     def verifyClass(self):
98         i=0
99         theClass=self.rw[i]
100         if((theClass>0) or (self.used<self.capacity)):
101             for j in range(self.capacity):
102                 if(self.rw[j]<theClass):
103                     theClass=self.rw[j]
104                     i=j
105                 if(theClass== -1):
106                     break
107             return i
108
109     '''does not uses class hierarchy so, just for comparing,
110 only occurs classes 2 (refenced and not modified) or 3
111 (refenced and modified), simulanting only read and write
112 situations'''

```

```

113 def method_LRU(self):
114     t1=time.time_ns()
115     for i in range (len(self.vMemory.pages)):
116
117         d=0
118         if(self.debug==1):
119             print("\n")
120             for e in range(len((self.frames))):
121                 if(self.frames[e]==self.vMemory.pages[i].address):
122                     print("#####")
123                     print("REQUESTED ALREADY IN RAM")
124                     d=1
125             print(self.vMemory.pages[i].address)
126             print(self.frames)
127
128         k=self.verifyOcurrence(self.vMemory.pages[i].address)
129         l=-1
130         if(k==-1):
131             self.frames.pop(0)
132             self.frames.append(self.vMemory.pages[i].address)
133             l=self.rw.pop(0)
134             self.pageFault+=1
135         else:
136             aux=self.frames.pop(k)
137             self.frames.append(self.vMemory.pages[i].address)
138             l=self.rw.pop(k)
139         if(self.vMemory.pages[i].action=='W'):
140             self.rw.append(3)
141         elif(self.vMemory.pages[i].action=='R'):
142             self.rw.append(2)
143         else:
144             self.rw.append(-1) #gone wrong
145         if((k==-1) and (l>=0) and (l<=3)):
146             if(l==0):
147                 self.nref+=1
148                 self.nmod+=1
149             elif(l==1):
150                 self.nref+=1
151                 self.mod+=1
152             elif(l==2):
153                 self.nmod+=1
154                 self.ref+=1
155             elif(l==3):
156                 self.mod+=1
157                 self.ref+=1
158
159         if(self.debug==1):
160             print(self.frames)
161             if(d==1):
162                 print("#####\n")
163     t2=time.time_ns()
164     print("\n## Least Recently Used (LRU) ##")
165     print("Page size: {}".format(self.vMemory.pageSize))
166     print("Physical memory: {}".format(self.size))
167     print("Physical memory page frames: {}".format(self.capacity))
168     print("Virtual memory: {}".format(self.vMemory.size))
169     print("Logical memory virtual pages: {}".format(self.vMemory.capacity))
170     print("Page faults: {}".format(self.pageFault))
171     print("Referenced (Replaced): {}".format(self.ref))
172     print("Unreferenced (Replaced): {}".format(self.nref))

```

```

173     print("Modified (Replaced): {}".format(self.mod))
174     print("Unmodified (Replaced): {}".format(self.nmod))
175     print("Time: {} seconds".format(((t2-t1)/(10**9))))
176     with open("lru4.txt", "a") as f:
177         f.write("## Least Recently Used (LRU) ##\n")
178         f.write("Page size: {}\n".format(self.vMemory.pageSize))
179         f.write("Physical memory: {}\n".format(self.size))
180         f.write("Physical memory page frames: {}\n".format(self.capacity))
181         f.write("Virtual memory: {}\n".format(self.vMemory.size))
182         f.write("Logical memory virtual pages: {}\n".format(self.vMemory.capacity))
183         f.write("Page faults: {}\n".format(self.pageFault))
184         f.write("Referenced (Replaced): {}\n".format(self.ref))
185         f.write("Unreferenced (Replaced): {}\n".format(self.nref))
186         f.write("Modified (Replaced): {}\n".format(self.mod))
187         f.write("Unmodified (Replaced): {}\n".format(self.nmod))
188         f.write("Time: {} seconds\n".format(((t2-t1)/(10**9))))
189     f.close()
190
191
192     def method_NRU(self):
193         clk1=int(len(self.frames)*2)
194         clk2=int(len(self.frames)*1.5)
195         clk3=int(len(self.frames)*1)
196         t1=time.time_ns()
197         for i in range(len(self.vMemory.pages)):
198
199             d=0
200             if(self.debug==1):
201                 print("\n")
202                 for e in range(len((self.frames))):
203                     if(self.frames[e]==self.vMemory.pages[i].address):
204                         print("#####")
205                         print("REQUESTED ALREADY IN RAM")
206                         d=1
207                 print(self.vMemory.pages[i].address)
208                 print(self.frames)
209                 print(self.rw)
210                 print(self.clock)
211
212             ''' CLOCK SIMULATION '''
213             for j in range(len(self.clock)):
214                 self.clock[j]+=1
215
216             ''' referenced and modified (stays for frames*2 clocks)
217                 referenced (stays for frames*1.5 clocks)
218                 modified (stays for frames clocks)
219             '''
220             for j in range(len(self.rw)):
221                 if(self.clock[j]==clk1 and self.rw[j]==3):
222                     self.rw[j]=1
223                     self.clock[j]=0
224                 elif(self.clock[j]==clk2 and self.rw[j]==2):
225                     self.rw[j]=0
226                     self.clock[j]=0
227                 elif(self.clock[j]==clk3 and self.rw[j]==1):
228                     self.rw[j]=0
229                     self.clock[j]=0
230
231             k=self.verifyOcurrency(self.vMemory.pages[i].address)
232             lowestClassIndex=-1

```

```

233         l=-1
234         if(k==-1):
235             lowestClassIndex=self.verifyClass()
236             self.frames.pop(lowestClassIndex)
237             self.frames.insert(lowestClassIndex, self.vMemory.pages[i].address)
238             l=self.rw[lowestClassIndex]
239             self.pageFault+=1
240             if(self.vMemory.pages[i].action=='W'):
241                 self.rw[lowestClassIndex]=3
242             elif(self.vMemory.pages[i].action=='R'):
243                 self.rw[lowestClassIndex]=2
244             else:
245                 self.rw[lowestClassIndex]=-1 #gone wrong
246             self.clock[lowestClassIndex]=0
247             if(self.used<self.capacity):
248                 self.used+=1
249         else:
250             if(self.vMemory.pages[i].action=='W'):
251                 self.rw[k]=3
252             elif(self.vMemory.pages[i].action=='R'):
253                 self.rw[k]=2
254             else:
255                 self.rw[k]=-1 #gone wrong
256             self.clock[k]=0
257         if((k==-1)and(l>=0) and (l<=3)):
258             if(l==0):
259                 self.nref+=1
260                 self.nmod+=1
261             elif(l==1):
262                 self.nref+=1
263                 self.mod+=1
264             elif(l==2):
265                 self.nmod+=1
266                 self.ref+=1
267             elif(l==3):
268                 self.mod+=1
269                 self.ref+=1
270
271         if(self.debug==1):
272             print(self.frames)
273             if(d==1):
274                 print("#####\n")
275         t2=time.time_ns()
276         print("\n## Not Recently Used (NRU) ##")
277         print("Page size: {}".format(self.vMemory.pageSize))
278         print("Physical memory: {}".format(self.size))
279         print("Virtual memory: {}".format(self.vMemory.size))
280         print("Physical memory page frames: {}".format(self.capacity))
281         print("Logical memory virtual pages: {}".format(self.vMemory.capacity))
282         print("Page faults: {}".format(self.pageFault))
283         print("Referenced (Replaced): {}".format(self.ref))
284         print("Unreferenced (Replaced): {}".format(self.nref))
285         print("Modified (Replaced): {}".format(self.mod))
286         print("Unmodified (Replaced): {}".format(self.nmod))
287         print("Time: {} seconds".format(((t2-t1)/(10**9))))
288         with open("nru3.txt", "a") as f:
289             f.write("## Not Recently Used (NRU) ##\n")
290             f.write("Page size: {}\n".format(self.vMemory.pageSize))
291             f.write("Physical memory: {}\n".format(self.size))
292             f.write("Physical memory page frames: {}\n".format(self.capacity))

```

```

293         f.write("Virtual memory: {}\n".format(self.vMemory.size))
294         f.write("Logical memory virtual pages: {}\n".format(self.vMemory.capacity))
295         f.write("Page faults: {}\n".format(self.pageFault))
296         f.write("Referenced (Replaced): {}\n".format(self.ref))
297         f.write("Unreferenced (Replaced): {}\n".format(self.nref))
298         f.write("Modified (Replaced): {}\n".format(self.mod))
299         f.write("Unmodified (Replaced): {}\n".format(self.nmod))
300         f.write("Time: {} seconds\n".format(((t2-t1)/(10**9))))
301     f.close()
302
303
304     def allocate(self, method):
305         if (method=="LRU"):
306             self.method_LRU()
307         elif (method=="NRU"):
308             self.method_NRU()
309
310
311
312     # "Processes" quantity generated considering logical memory size
313     # and page size
314     # Consider main value in kB
315
316     if __name__ == "__main__":
317         debug=0
318         page_size=64
319         vmem=vMemory(1024000, page_size)
320         vmem.generate()
321         ram=RAM(128000, vmem, page_size, debug)
322         ram.setUpRAM()
323         ram.allocate("NRU")

```

## Apêndice A2 - Algoritmo para obtenção das médias

```

1 t="nru4.txt"
2
3 with open(t, "r") as f:
4     lines=f.readlines()
5 k=0
6
7 string = [" " for x in range(48)]
8 string2 = [" " for x in range(48)]
9
10 for i in range (4):
11     for j in range(12):
12         string[j+k]=' '.join(lines[k+j])
13     k+=12
14
15 for i in range (48):
16     if (i!=0 and i!=12 and i!=24 and i!=36):
17         string[i]=string[i].strip().split(":")
18         string2[i]=string[i][1]
19         if (i==11 or i==23 or i==35 or i==47):
20             print(string2[i])
21             string2[i]=string2[i][: -8]
22             string2[i]=string2[i].strip()
23
24 print(string2)
25 with open(t, "a") as f:
26     f.write("\nAverage\n")

```

```
27     for i in range (11):
28         #f.write(string[i+1][0])
29         k=(float(string2[i+1])+float(string2[i+13])+float(string2[i+25])+float(string2[i
30         +37]))/4
31         #print(k)
32         f.write(string[i+1][0]+ ": "+str(k)+"\n")
33     f.close()
34 #print(string2)
35 #print(string)
```

## Apêndice B - Resultados

Os resultados são mostrados já com a respectiva média.

### Apêndice B1 - Teste 1 (LRU)

```
## Least Recently Used (LRU) ##
Page size: 4
Physical memory: 128
Physical memory page frames: 32
Virtual memory: 1024
Logical memory virtual pages: 256
Page faults: 235
Referenced (Replaced): 203
Unreferenced (Replaced): 0
Modified (Replaced): 55
Unmodified (Replaced): 148
Time: 0.0019061 seconds
## Least Recently Used (LRU) ##
Page size: 4
Physical memory: 128
Physical memory page frames: 32
Virtual memory: 1024
Logical memory virtual pages: 256
Page faults: 242
Referenced (Replaced): 210
Unreferenced (Replaced): 0
Modified (Replaced): 51
Unmodified (Replaced): 159
Time: 0.0019331 seconds
## Least Recently Used (LRU) ##
Page size: 4
Physical memory: 128
Physical memory page frames: 32
Virtual memory: 1024
Logical memory virtual pages: 256
Page faults: 236
Referenced (Replaced): 204
Unreferenced (Replaced): 0
Modified (Replaced): 53
Unmodified (Replaced): 151
Time: 0.0019311 seconds
## Least Recently Used (LRU) ##
Page size: 4
Physical memory: 128
Physical memory page frames: 32
Virtual memory: 1024
Logical memory virtual pages: 256
Page faults: 243
Referenced (Replaced): 211
Unreferenced (Replaced): 0
Modified (Replaced): 52
Unmodified (Replaced): 159
Time: 0.0019337 seconds

Average
Page size: 4.0
Physical memory: 128.0
```

Physical memory page frames: 32.0  
Virtual memory: 1024.0  
Logical memory virtual pages: 256.0  
Page faults: 239.0  
Referenced (Replaced): 207.0  
Unreferenced (Replaced): 0.0  
Modified (Replaced): 52.75  
Unmodified (Replaced): 154.25  
Time: 0.001926

## Apêndice B2 - Teste 2 (LRU)

```
## Least Recently Used (LRU) ##  
Page size: 8  
Physical memory: 128  
Physical memory page frames: 16  
Virtual memory: 1024  
Logical memory virtual pages: 128  
Page faults: 120  
Referenced (Replaced): 104  
Unreferenced (Replaced): 0  
Modified (Replaced): 28  
Unmodified (Replaced): 76  
Time: 0.0009048 seconds  
## Least Recently Used (LRU) ##  
Page size: 8  
Physical memory: 128  
Physical memory page frames: 16  
Virtual memory: 1024  
Logical memory virtual pages: 128  
Page faults: 121  
Referenced (Replaced): 105  
Unreferenced (Replaced): 0  
Modified (Replaced): 24  
Unmodified (Replaced): 81  
Time: 0.0009361 seconds  
## Least Recently Used (LRU) ##  
Page size: 8  
Physical memory: 128  
Physical memory page frames: 16  
Virtual memory: 1024  
Logical memory virtual pages: 128  
Page faults: 118  
Referenced (Replaced): 102  
Unreferenced (Replaced): 0  
Modified (Replaced): 24  
Unmodified (Replaced): 78  
Time: 0.0009321 seconds  
## Least Recently Used (LRU) ##  
Page size: 8  
Physical memory: 128  
Physical memory page frames: 16  
Virtual memory: 1024  
Logical memory virtual pages: 128  
Page faults: 119  
Referenced (Replaced): 103  
Unreferenced (Replaced): 0  
Modified (Replaced): 26  
Unmodified (Replaced): 77  
Time: 0.0 seconds
```



Average  
Page size: 8.0  
Physical memory: 128.0  
Physical memory page frames: 16.0  
Virtual memory: 1024.0  
Logical memory virtual pages: 128.0  
Page faults: 119.5  
Referenced (Replaced): 103.5  
Unreferenced (Replaced): 0.0  
Modified (Replaced): 25.5  
Unmodified (Replaced): 78.0  
Time: 0.00069325

## Apêndice B3 - Teste 3 (LRU)

## Least Recently Used (LRU) ##  
Page size: 32  
Physical memory: 128000  
Physical memory page frames: 4000  
Virtual memory: 1024000  
Logical memory virtual pages: 32000  
Page faults: 29152  
Referenced (Replaced): 25152  
Unreferenced (Replaced): 0  
Modified (Replaced): 6312  
Unmodified (Replaced): 18840  
Time: 16.1398173 seconds  
## Least Recently Used (LRU) ##  
Page size: 32  
Physical memory: 128000  
Physical memory page frames: 4000  
Virtual memory: 1024000  
Logical memory virtual pages: 32000  
Page faults: 29201  
Referenced (Replaced): 25201  
Unreferenced (Replaced): 0  
Modified (Replaced): 6261  
Unmodified (Replaced): 18940  
Time: 15.9324246 seconds  
## Least Recently Used (LRU) ##  
Page size: 32  
Physical memory: 128000  
Physical memory page frames: 4000  
Virtual memory: 1024000  
Logical memory virtual pages: 32000  
Page faults: 29152  
Referenced (Replaced): 25152  
Unreferenced (Replaced): 0  
Modified (Replaced): 6247  
Unmodified (Replaced): 18905  
Time: 15.780833 seconds  
## Least Recently Used (LRU) ##  
Page size: 32  
Physical memory: 128000  
Physical memory page frames: 4000  
Virtual memory: 1024000  
Logical memory virtual pages: 32000  
Page faults: 29153  
Referenced (Replaced): 25153

Unreferenced (Replaced): 0  
Modified (Replaced): 6166  
Unmodified (Replaced): 18987  
Time: 15.7150088 seconds

#### Average

Page size: 32.0  
Physical memory: 128000.0  
Physical memory page frames: 4000.0  
Virtual memory: 1024000.0  
Logical memory virtual pages: 32000.0  
Page faults: 29164.5  
Referenced (Replaced): 25164.5  
Unreferenced (Replaced): 0.0  
Modified (Replaced): 6246.5  
Unmodified (Replaced): 18918.0  
Time: 15.892020925

### Apêndice B4 - Teste 4 (LRU)

#### ## Least Recently Used (LRU) ##

Page size: 64  
Physical memory: 128000  
Physical memory page frames: 2000  
Virtual memory: 1024000  
Logical memory virtual pages: 16000  
Page faults: 14593  
Referenced (Replaced): 12593  
Unreferenced (Replaced): 0  
Modified (Replaced): 3244  
Unmodified (Replaced): 9349  
Time: 3.8626708 seconds

#### ## Least Recently Used (LRU) ##

Page size: 64  
Physical memory: 128000  
Physical memory page frames: 2000  
Virtual memory: 1024000  
Logical memory virtual pages: 16000  
Page faults: 14552  
Referenced (Replaced): 12552  
Unreferenced (Replaced): 0  
Modified (Replaced): 3069  
Unmodified (Replaced): 9483  
Time: 3.919519 seconds

#### ## Least Recently Used (LRU) ##

Page size: 64  
Physical memory: 128000  
Physical memory page frames: 2000  
Virtual memory: 1024000  
Logical memory virtual pages: 16000  
Page faults: 14560  
Referenced (Replaced): 12560  
Unreferenced (Replaced): 0  
Modified (Replaced): 3136  
Unmodified (Replaced): 9424  
Time: 3.9843462 seconds

#### ## Least Recently Used (LRU) ##

Page size: 64  
Physical memory: 128000  
Physical memory page frames: 2000

Virtual memory: 1024000  
Logical memory virtual pages: 16000  
Page faults: 14600  
Referenced (Replaced): 12600  
Unreferenced (Replaced): 0  
Modified (Replaced): 3105  
Unmodified (Replaced): 9495  
Time: 4.0103057 seconds

Average  
Page size: 64.0  
Physical memory: 128000.0  
Physical memory page frames: 2000.0  
Virtual memory: 1024000.0  
Logical memory virtual pages: 16000.0  
Page faults: 14576.25  
Referenced (Replaced): 12576.25  
Unreferenced (Replaced): 0.0  
Modified (Replaced): 3138.5  
Unmodified (Replaced): 9437.75  
Time: 3.944210425

## Apêndice B5 - Teste 1 (NRU)

## Not Recently Used (NRU) ##  
Page size: 4  
Physical memory: 128  
Physical memory page frames: 32  
Virtual memory: 1024  
Logical memory virtual pages: 256  
Page faults: 234  
Referenced (Replaced): 97  
Unreferenced (Replaced): 105  
Modified (Replaced): 38  
Unmodified (Replaced): 164  
Time: 0.0040067 seconds  
## Not Recently Used (NRU) ##  
Page size: 4  
Physical memory: 128  
Physical memory page frames: 32  
Virtual memory: 1024  
Logical memory virtual pages: 256  
Page faults: 234  
Referenced (Replaced): 101  
Unreferenced (Replaced): 101  
Modified (Replaced): 40  
Unmodified (Replaced): 162  
Time: 0.0040189 seconds  
## Not Recently Used (NRU) ##  
Page size: 4  
Physical memory: 128  
Physical memory page frames: 32  
Virtual memory: 1024  
Logical memory virtual pages: 256  
Page faults: 235  
Referenced (Replaced): 100  
Unreferenced (Replaced): 103  
Modified (Replaced): 42  
Unmodified (Replaced): 161  
Time: 0.0039911 seconds

## Not Recently Used (NRU) ##

Page size: 4  
Physical memory: 128  
Physical memory page frames: 32  
Virtual memory: 1024  
Logical memory virtual pages: 256  
Page faults: 232  
Referenced (Replaced): 103  
Unreferenced (Replaced): 97  
Modified (Replaced): 42  
Unmodified (Replaced): 158  
Time: 0.0040169 seconds

Average

Page size: 4.0  
Physical memory: 128.0  
Physical memory page frames: 32.0  
Virtual memory: 1024.0  
Logical memory virtual pages: 256.0  
Page faults: 233.75  
Referenced (Replaced): 100.25  
Unreferenced (Replaced): 101.5  
Modified (Replaced): 40.5  
Unmodified (Replaced): 161.25  
Time: 0.004008400000000005

## Apêndice B6 - Teste 2 (NRU)

## Not Recently Used (NRU) ##

Page size: 8  
Physical memory: 128  
Physical memory page frames: 16  
Virtual memory: 1024  
Logical memory virtual pages: 128  
Page faults: 120  
Referenced (Replaced): 54  
Unreferenced (Replaced): 50  
Modified (Replaced): 16  
Unmodified (Replaced): 88  
Time: 0.0009989 seconds

## Not Recently Used (NRU) ##

Page size: 8  
Physical memory: 128  
Physical memory page frames: 16  
Virtual memory: 1024  
Logical memory virtual pages: 128  
Page faults: 117  
Referenced (Replaced): 50  
Unreferenced (Replaced): 51  
Modified (Replaced): 18  
Unmodified (Replaced): 83  
Time: 0.0009327 seconds

## Not Recently Used (NRU) ##

Page size: 8  
Physical memory: 128  
Physical memory page frames: 16  
Virtual memory: 1024  
Logical memory virtual pages: 128  
Page faults: 118  
Referenced (Replaced): 53

Unreferenced (Replaced): 49  
Modified (Replaced): 24  
Unmodified (Replaced): 78  
Time: 0.0009335 seconds  
## Not Recently Used (NRU) ##  
Page size: 8  
Physical memory: 128  
Physical memory page frames: 16  
Virtual memory: 1024  
Logical memory virtual pages: 128  
Page faults: 121  
Referenced (Replaced): 54  
Unreferenced (Replaced): 51  
Modified (Replaced): 21  
Unmodified (Replaced): 84  
Time: 0.0019061 seconds

Average  
Page size: 8.0  
Physical memory: 128.0  
Physical memory page frames: 16.0  
Virtual memory: 1024.0  
Logical memory virtual pages: 128.0  
Page faults: 119.0  
Referenced (Replaced): 52.75  
Unreferenced (Replaced): 50.25  
Modified (Replaced): 19.75  
Unmodified (Replaced): 83.25  
Time: 0.0011928

## Apêndice B7 - Teste 3 (NRU)

## Not Recently Used (NRU) ##  
Page size: 32  
Physical memory: 128000  
Physical memory page frames: 4000  
Virtual memory: 1024000  
Logical memory virtual pages: 32000  
Page faults: 29089  
Referenced (Replaced): 11379  
Unreferenced (Replaced): 13710  
Modified (Replaced): 4915  
Unmodified (Replaced): 20174  
Time: 62.4230931 seconds  
## Not Recently Used (NRU) ##  
Page size: 32  
Physical memory: 128000  
Physical memory page frames: 4000  
Virtual memory: 1024000  
Logical memory virtual pages: 32000  
Page faults: 29086  
Referenced (Replaced): 11457  
Unreferenced (Replaced): 13629  
Modified (Replaced): 4984  
Unmodified (Replaced): 20102  
Time: 63.394496 seconds  
## Not Recently Used (NRU) ##  
Page size: 32  
Physical memory: 128000  
Physical memory page frames: 4000

Virtual memory: 1024000  
Logical memory virtual pages: 32000  
Page faults: 29180  
Referenced (Replaced): 11479  
Unreferenced (Replaced): 13701  
Modified (Replaced): 4987  
Unmodified (Replaced): 20193  
Time: 63.226975 seconds  
## Not Recently Used (NRU) ##  
Page size: 32  
Physical memory: 128000  
Physical memory page frames: 4000  
Virtual memory: 1024000  
Logical memory virtual pages: 32000  
Page faults: 29193  
Referenced (Replaced): 11518  
Unreferenced (Replaced): 13675  
Modified (Replaced): 5087  
Unmodified (Replaced): 20106  
Time: 62.5876528 seconds

Average  
Page size: 32.0  
Physical memory: 128000.0  
Physical memory page frames: 4000.0  
Virtual memory: 1024000.0  
Logical memory virtual pages: 32000.0  
Page faults: 29137.0  
Referenced (Replaced): 11458.25  
Unreferenced (Replaced): 13678.75  
Modified (Replaced): 4993.25  
Unmodified (Replaced): 20143.75  
Time: 62.908054225

## Apêndice B8 - Teste 4 (NRU)

## Not Recently Used (NRU) ##  
Page size: 64  
Physical memory: 128000  
Physical memory page frames: 2000  
Virtual memory: 1024000  
Logical memory virtual pages: 16000  
Page faults: 14561  
Referenced (Replaced): 5774  
Unreferenced (Replaced): 6787  
Modified (Replaced): 2514  
Unmodified (Replaced): 10047  
Time: 15.9333692 seconds  
## Not Recently Used (NRU) ##  
Page size: 64  
Physical memory: 128000  
Physical memory page frames: 2000  
Virtual memory: 1024000  
Logical memory virtual pages: 16000  
Page faults: 14492  
Referenced (Replaced): 5652  
Unreferenced (Replaced): 6840  
Modified (Replaced): 2486  
Unmodified (Replaced): 10006  
Time: 15.8017487 seconds

```
## Not Recently Used (NRU) ##
Page size: 64
Physical memory: 128000
Physical memory page frames: 2000
Virtual memory: 1024000
Logical memory virtual pages: 16000
Page faults: 14584
Referenced (Replaced): 5761
Unreferenced (Replaced): 6823
Modified (Replaced): 2541
Unmodified (Replaced): 10043
Time: 15.8635839 seconds
## Not Recently Used (NRU) ##
Page size: 64
Physical memory: 128000
Physical memory page frames: 2000
Virtual memory: 1024000
Logical memory virtual pages: 16000
Page faults: 14681
Referenced (Replaced): 5814
Unreferenced (Replaced): 6867
Modified (Replaced): 2495
Unmodified (Replaced): 10186
Time: 15.4696374 seconds
```

#### Average

```
Page size: 64.0
Physical memory: 128000.0
Physical memory page frames: 2000.0
Virtual memory: 1024000.0
Logical memory virtual pages: 16000.0
Page faults: 14579.5
Referenced (Replaced): 5750.25
Unreferenced (Replaced): 6829.25
Modified (Replaced): 2509.0
Unmodified (Replaced): 10070.5
Time: 15.7670848
```