

# MAC0422 - Sistemas Operacionais

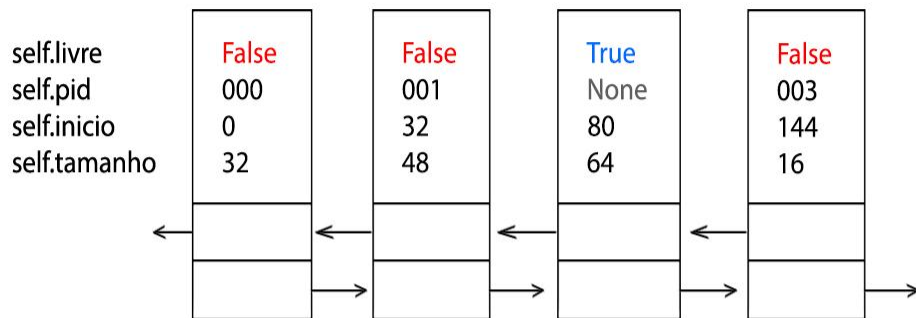
EP2 - Simulador de gerência de memória

Caio Lopes - NUSP: 7991187

Marcos K Yamazaki - NUSP: 7577622

# Estrutura de dados

**memoriaVirtual** com 160 bytes, ou seja temos 10 páginas



-> Referência para  
o Page Frame da  
memória física  
-> 255 significa  
que não está lá

**mapeamento** = []

Processo -> id:000 tamanho: 30 bytes

Ocupa 2 paginas na memória

Processo -> id:001 tamanho: 48 bytes

Ocupa 3 páginas interias

Processo -> id:003 tamanho: 10 bytes

Ocupa apenas 1 página

**mapeamento**[0] = 255

**mapeamento**[1] = 001

**mapeamento**[2] = 000

**mapeamento**[3] = 255

**mapeamento**[4] = 255

mapeamento[5] = 255

mapeamento[6] = 255

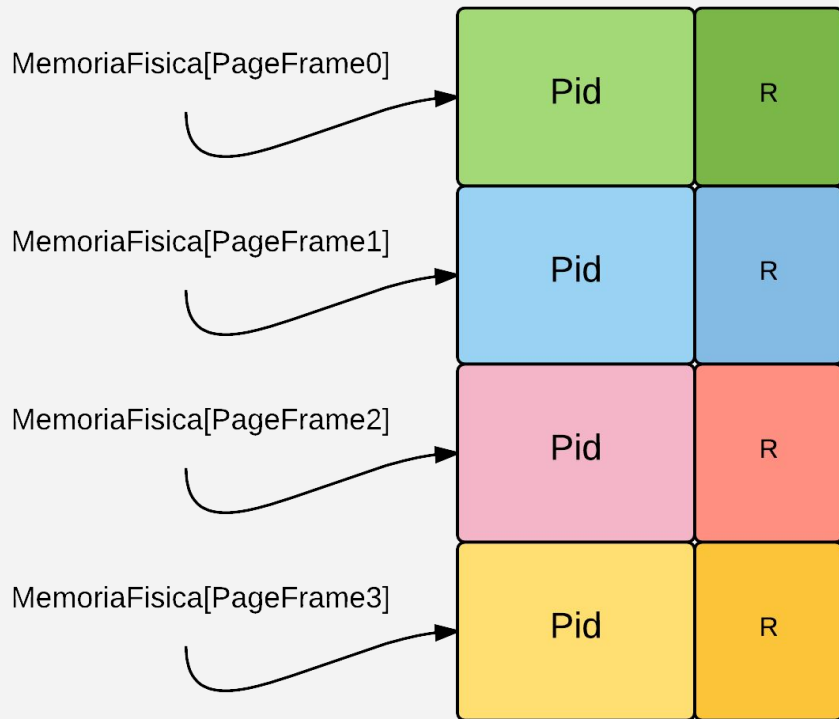
mapeamento[7] = 255

mapeamento[8] = 255

**mapeamento**[9] = 003

# Estrutura de dados

## Memória Física

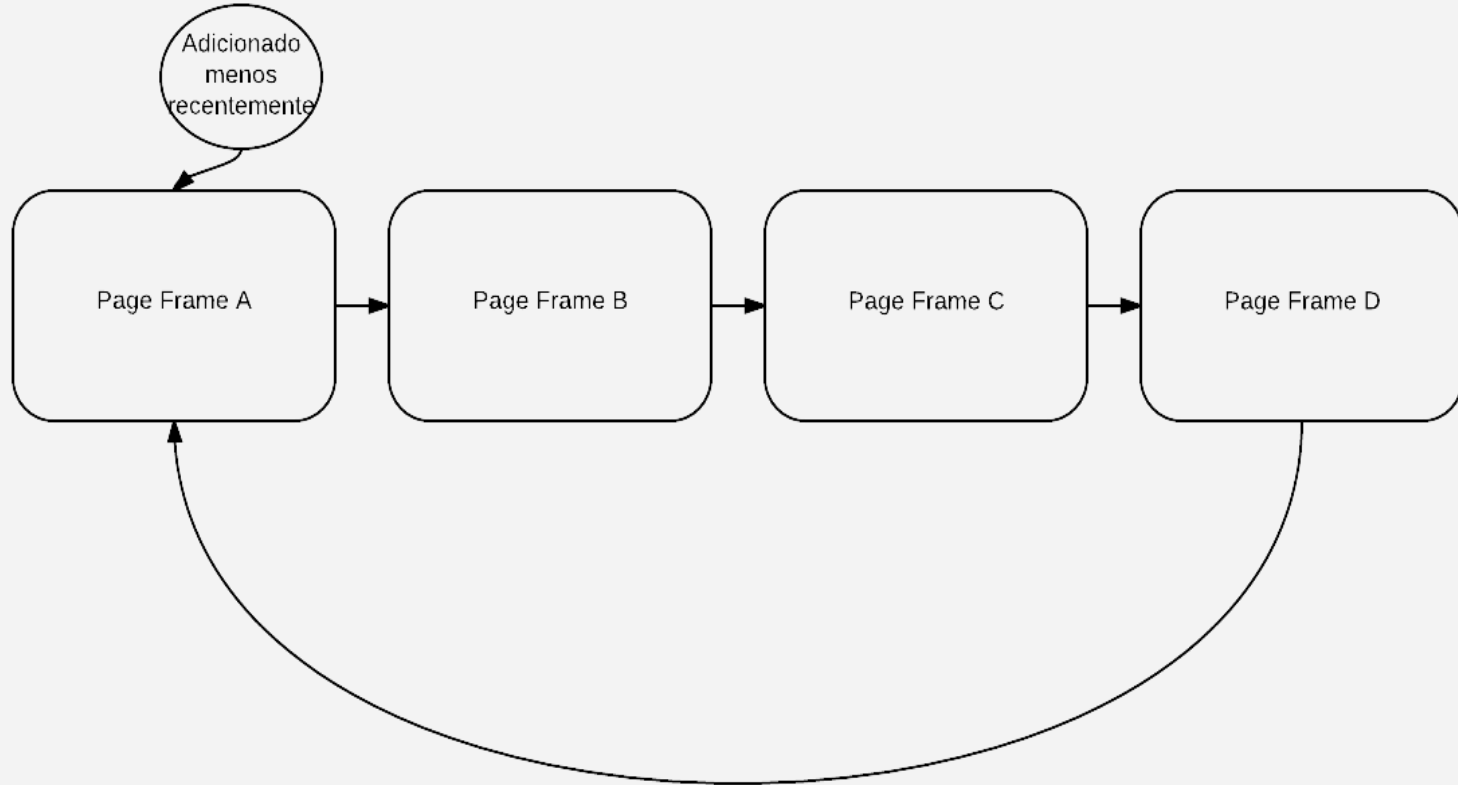


Cada pageFrame esta associado a um PID do processo que está utilizando aquele setor da memoria física e o bit R que determina se aquele pageFrame foi acessado recentemente

Neste exercício-programa, resetamos os bits R de todas as page frames a cada 5 segundos

# Estrutura de dados

FIFO



# Simulações

48 96

0 processoA 7 32 0 2 18 5

1 processoB 3 48 32 2

4 processoC 7 16 0 6

## First Fit vs Next Fit

Memoria Virtual:

Tempo de execução:							
pagina	pid	0	1	2	3	4	
000	255	000	000	000	000	000	
001	255	000	000	000	000	000	
002	255	255	001	001	255	002	
003	255	255	001	001	255	255	
004	255	255	001	001	255	255	
005	255	255	255	255	255	255	

Tempo de execução:							
pagina	pid	0	1	2	3	4	
000	255	000	000	000	000	000	
001	255	000	000	000	000	000	
002	255	255	001	001	255	255	
003	255	255	001	001	255	255	
004	255	255	001	001	255	255	
005	255	255	255	255	255	002	

Seria melhor usar o Next Fit, quando tem menos processos que acabam antes de outros processos chegarem a CPU, pois o First Fit percorre toda a memória virtual até achar um espaço livre.

# Justiça Not Recently Used Page

48 256

0 processoA 9 20 17 1 18 4

2 processoB 11 20 19 2

3 processoC 10 40 0 3 17 6

Memoria Fisica				Memoria Fisica				Memoria Fisica				Memoria Fisica				Memoria Fisica				Memoria Fisica			
+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+			
Page Frame   pid   R				Page Frame   pid   R				Page Frame   pid   R				Page Frame   pid   R				Page Frame   pid   R				Page Frame   pid   R			
+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+			
000   000   1				000   000   1				000   000   1				000   000   1				000   000   0				000   002   1			
001   255   0				001   001   1				001   001   1				001   001   1				001   001   0				001   001   0			
002   255   0				002   255   0				002   002   1				002   002   1				002   002   0				002   002   0			
+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+				+-----+-----+-----+			

PageFrame0 é referenciado inicialmente, seguido por pageFrame 1 e 2

PageFrame 0 é referenciado novamente, sendo agora o ultimo referenciado

Todos os R são zerados no tempo 5

Há pageFault quando o processoC tenta acessar a memoria, e o pageFrame que acabou de ser utilizado, e foi referenciado duas vezes, é trocado.

# Justiça First-In First-out

48 96

0 processoA 10 32 0 1 0 6  
1 processoB 9 48 32 2 16 7  
4 processoC 12 16 0 5 10 7

FIFO funcionou, porém, neste caso, quando deu Page Fault, ela retirou um quadro de página que um segundo atrás ela tinha acabado de ser referenciada. O que não seria uma algo muito esperto.

Tempo de execucao: 6

Memoria Virtual

pagina	pid	address
000	000	000
001	000	255
002	001	255
003	001	255
004	001	001
005	002	002

Memoria Fisica

Page Frame	pid	R
000	000	1
001	001	0
002	002	0

Tempo de execucao: 7

Memoria Virtual

pagina	pid	address
000	000	255
001	000	255
002	001	255
003	001	000
004	001	001
005	002	002

Memoria Fisica

Page Frame	pid	R
000	001	1
001	001	0
002	002	0

# Justiça Second Chance Page

48 96

0 processoA 10 32 0 1 0 3

1 processoB 9 48 32 2 16 4

2 processoC 12 16 0 3 10 7

Também funciona de acordo com o algoritmo, porém, neste caso foi pior que o FIFO, já que além de ter percorrido por todas os page frames zerando o bit R, o escolhido foi a primeira frame, que ainda assim não foi uma boa escolha. (Pequenas chances de isso acontecer)

Tempo de execucao: 3

Memoria Virtual

pagina	pid	address
000	000	000
001	000	255
002	001	255
003	001	255
004	001	001
005	002	002

Memoria Fisica

Page Frame	pid	R
000	000	1
001	001	1
002	002	1

Tempo de execucao: 4

Memoria Virtual

pagina	pid	address
000	000	255
001	000	255
002	001	255
003	001	000
004	001	001
005	002	002

Memoria Fisica

Page Frame	pid	R
000	001	1
001	001	0
002	002	0



# Justiça Least Recently Used

48 96

0 processoA 10 32 0 1 0 3  
 1 processoB 9 48 32 2 16 4  
 2 processoC 12 16 0 3 10 7

Usando o mesmo exemplo da anterior, de fato aqui conseguimos retirar a page frame que foi acessada menos recentemente.

Não podemos dizer que ela será a mais justa, pois não temos como adivinhar os acessos as páginas futuras.

Tempo de execucao: 3

Memoria Virtual

pagina	pid	address
000	000	000
001	000	255
002	001	255
003	001	255
004	001	001
005	002	002

Memoria Fisica

Page	Frame	pid	R
000	000	1	
001	001	1	
002	002	1	

Tempo de execucao: 4

Memoria Virtual

pagina	pid	address
000	000	000
001	000	255
002	001	255
003	001	001
004	001	255
005	002	002

Memoria Fisica

Page	Frame	pid	R
000	000	1	
001	001	1	
002	002	1	

-> Manteve a página 000 no quadro de página 000

-> Trocou o quadro de pagina 001, que foi a menos recentemente usada, da página 4 para o 3

-> Pid não altera na memória física, já que as páginas 2, 3 e 4 são dos mesmo processos