



## **GERÊNCIA DE MEMÓRIA VIRTUAL**

DISCENTE:  
TEÓFILO VITOR DE CARVALHO CLEMENTE | 20220080516

DOCENTE:  
DIOGO PINHEIRO FERNANDES PEDROSA



## 1. INTRODUÇÃO

A presente atividade visa apresentar a resolução dos exercícios sobre gerenciamento de memória virtual, onde serão apresentados os programas usados para obter as respostas para os problemas e as respectivas respostas. Desse modo, aplicando de forma prática o aprendizado adquirido sobre o assunto.

## 2. METODOLOGIA

Para a realização destes exercícios foram utilizadas pesquisas no material didático fornecido pelo professor em vídeo e posterior desenvolvimento dos problemas utilizando os códigos em python como veremos a seguir.

## 3. CÓDIGOS

Nesta seção serão apresentados os códigos em python utilizados para resolução das questões.

### 3.1 ALGORITMO FIFO

```
#FIFO

def FIFO (numquadros,referencias):

    cont=1

    quadros = np.full(numquadros, np.inf)

    numacertos = 0

    numfalhas = 0

    for i in range(0,len(referencias)): #for para percorrer o processo

        if referencias[i] in quadros: #se for encontrado em quadros

            numacertos +=1 #computa os acertos

        else: #caso não seja encontrado

            numfalhas += 1 #computa as falhas

            quadros[cont] = referencias[i] #comparação vet quadros | vet referencias

            cont+=1 # acrescimo no contador
```



```
if cont >= numquadros:

    cont = 0 #zera o contador caso exceda o tamanho de quadros

print("Processo ref: ", referencias) #vetor referencias impressão
print("ALGORITMO FIFO:") #impressão respostas
print("Acertos: ", numacertos)
print("Falhas: ", numfalhas)
```

### 3.2 ALGORITMO LRU

#LRU

```
def LRU(numquadros, referencias):

    numacertos = 0

    numfalhas = 0

    paginas = []

    for k in referencias:

        if k in paginas: #caso já exista a pagina é removida e adicionada no fim

            paginas.remove(k) #remove

            paginas.append(k) #adiciona

            numacertos += 1 #quantidade de acessos

        else: #quando não foi dectada a página

            numfalhas +=1 #quantidade de falhas

            if(len(paginas) < numquadros): #verifica o tamanho da lista de páginas

                paginas.append(k) #adiciona

            else: #caso o tamanho seja maior que a memória

                paginas.remove(paginas[0]) #remove

                paginas.append(k) #adiciona
```



```
print("ALGORITMO LRU:") #impressão respostas

print("Acertos: ", numacertos)

print("Falhas: ", numfalhas)
```

### 3.3 ALGORITMO ÓTIMO

```
#OTIMO
```

```
def OTIMO(numquadros,referencias):
```

```
    quadros = []
```

```
    numacertos = 0
```

```
    numfalhas = 0
```

```
    init = 'No'
```

```
    occurance = [None for i in range(numquadros)]
```

```
    for i in range(len(referencias)): #percorre referencias
```

```
        if referencias[i] not in quadros: #caso não esteja em quadros
```

```
            if len(quadros)<numquadros: #caso seja menor que numquadros
```

```
                quadros.append(referencias[i]) #adiciona
```

```
            else:
```

```
                for x in range(len(quadros)): #percorre quadros
```

```
                    if quadros[x] not in referencias[i+1:]: #em relação ao elemento
posterior
```

```
                        quadros[x] = referencias[i] #comparação
```

```
                        break
```

```
                else:
```

```
                    occurance[x] = referencias[i+1:].index(quadros[x])
```

```
            else:
```



```
quadros[occurrence.index(max(occurrence))] = referencias[i]

numfalhas += 1 #computa as falhas

init = 'Falha'

else:

    init = 'Acerto'

    numacertos +=1 #computa os acertos

print("ALGORITMO ÓTIMO:") #impressão respostas

print("Acertos: ", numacertos)

print("Falhas: ", numfalhas)
```

## 4. RESULTADOS

A seguir serão apresentados os resultados de cada algoritmo para três, quatro, cinco, seis e sete quadros respectivamente.

### 4.1 TRÊS QUADROS

```
Processo ref: [1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 6]
ALGORITMO FIFO:
Acertos: 5
Falhas: 14
ALGORITMO LRU:
Acertos: 4
Falhas: 15
ALGORITMO ÓTIMO:
Acertos: 9
Falhas: 10
```

Figura 1 - Resultado para 3 quadros.

## 4.2 QUATRO QUADROS

```
Processo ref: [1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 6]
ALGORITMO FIFO:
Acertos: 6
Falhas: 13
ALGORITMO LRU:
Acertos: 9
Falhas: 10
ALGORITMO ÓTIMO:
Acertos: 11
Falhas: 8
```

Figura 2 - Resultado para 4 quadros.

## 4.3 CINCO QUADROS

```
Processo ref: [1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 6]
ALGORITMO FIFO:
Acertos: 9
Falhas: 10
ALGORITMO LRU:
Acertos: 11
Falhas: 8
ALGORITMO ÓTIMO:
Acertos: 12
Falhas: 7
```

Figura 3 - Resultado para 5 quadros.

## 4.4 SEIS QUADROS

```
Processo ref: [1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 6]
ALGORITMO FIFO:
Acertos: 10
Falhas: 9
ALGORITMO LRU:
Acertos: 12
Falhas: 7
ALGORITMO ÓTIMO:
Acertos: 12
Falhas: 7
```

Figura 4 - Resultado para 6 quadros.



## 4.5 SETE QUADROS

```
Processo ref: [1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 6]
ALGORITMO FIFO:
Acertos: 12
Falhas: 7
ALGORITMO LRU:
Acertos: 12
Falhas: 7
ALGORITMO ÓTIMO:
Acertos: 12
Falhas: 7
```

Figura 5 - Resultado para 7 quadros.

## 5. CONCLUSÕES

A partir do estudo e desenvolvimento dos exercícios sobre gerenciamento de memória virtual foi possível agregar conhecimento ao adquirido vendo os vídeos do professor e realizar de forma prática a teoria vista, aplicando assim tal conhecimento.

## 6. REFERÊNCIAS

- [1] SILBERSCHATZ, A., GALVIN, P.B., GAGNE, G. Fundamentos de sistemas operacionais, Ed. LTC, 8ª ed., 2011.
- [2] Material didático em vídeo do Professor Diogo.
- [3] Google Colaboratory. Disponível em: <https://colab.research.google.com/>.