OS | Assignment 3

operating systems

by Soravis Varayuththasawad 5913094

Page replacemnet

Page replacement

FIFO

LRU

Optimal

Input and common function

```
print("Select the method (1 , 2, 3) : \n")
print("1. FIFO")
print("2. LRU")
print("3. OPT")
select = input()

page_add = input()
page_add = page_add.split()

for i in range(len(page_add)):
    page_add[i] = eval(page_add[i])

frame = input()
frame = eval(frame)

allocation = [[0 for i in range(frame)] for j in range(len(page_add)) ]

print("--------------------------------------------------------------------------------")

def isFull(x):
    for i in range(frame):
        if allocation[x][i] == 0:
            return False
    return True

def isDuplicate(fm, key):
    for i in range(frame):
        if allocation[fm][i] == key:
            return True
    return False
```

Input sample

```
Select the method (1 , 2, 3) :

1. FIFO
2. LRU
3. OPT
1
2 3 2 1 5 2 4 5 3 2 5 2
3
```

Common piece of code

```
page_fault = 0
for i in range(len(page_add)):
    if i >= 1:
        for n in range(frame):
            allocation[i][n] = allocation[i-1][n]

    if isDuplicate(i,page_add[i]) == False:
        if isFull(i) == False:
            for j in range(frame):
                if allocation[i][j] == 0:
                    allocation[i][j] = page_add[i]
                    break
        else:
```

FIFO

Page address stream

2 3 2 1 5 2 4 5 3 2 5 2

FIFO

FIFO

```
count = [0 for m in range(frame)]
for j in range(frame):
    target = allocation[i][j]
    mx = -1
    for k in range(i,0,-1):
        if allocation[k][j] == target:
            count[j] = count[j] + 1
        else:
            break
mx = -1
for n in range(len(count)):
    mx = max(mx, count[n])
allocation[i][count.index(mx)] = page_add[i]
page_fault = page_fault + 1
```

FIFO

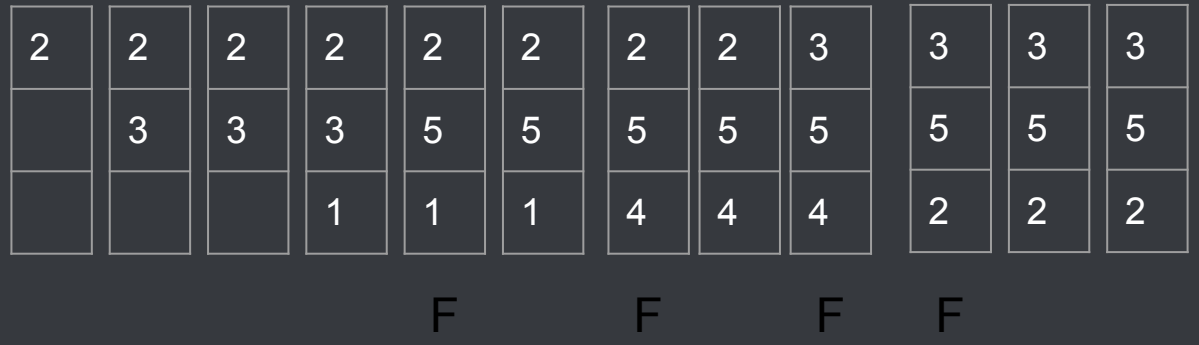
```
Select the method (1 , 2, 3) :

1. FIFO
2. LRU
3. OPT
1
2 3 2 1 5 2 4 5 3 2 5 2
3
--------------------------------------------------------------------------
page fault : 6
[[2, 0, 0], [2, 3, 0], [2, 3, 0], [2, 3, 1], [5, 3, 1], [5, 2, 1], [5, 2, 4], [5, 2, 4], [3, 2, 4], [3, 2, 4], [3, 5, 4], [3, 5, 2]]
```

LRU

Page address stream

2 3 2 1 5 2 4 5 3 2 5 2

LRU



LRU

```
count = [0 for m in range(frame)]
for j in range(frame):
    target = allocation[i][j]
    mx = -1
    for k in range(i,0,-1):
        if page_add[k] != target:
            count[j] = count[j] + 1
        else:
            break
mx = -1
for n in range(len(count)):
    mx = max(mx, count[n])
allocation[i][count.index(mx)] = page_add[i]
page_fault = page_fault + 1
```

LRU

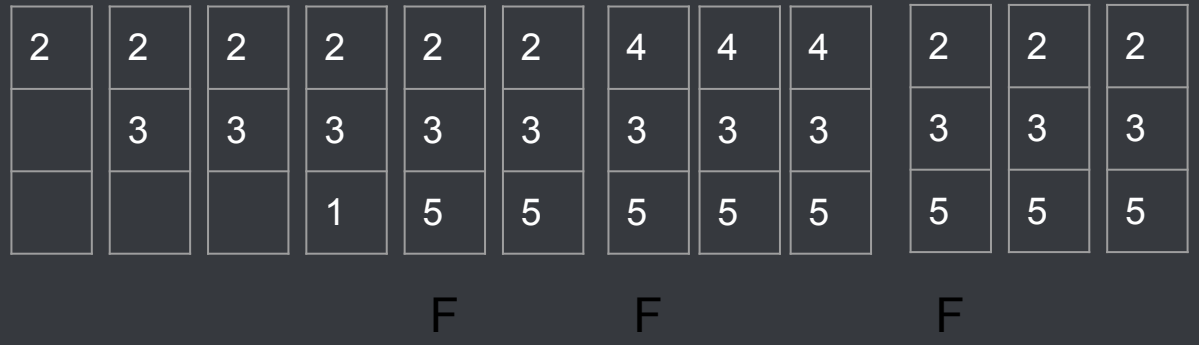
```
Select the method (1 , 2, 3) :

1. FIFO
2. LRU
3. OPT
2
2 3 2 1 5 2 4 5 3 2 5 2
3
--------------------------------------------------------------------------
page fault : 4
[[2, 0, 0], [2, 3, 0], [2, 3, 0], [2, 3, 1], [2, 5, 1], [2, 5, 1], [2, 5, 4], [2, 5, 4], [3, 5, 4], [3, 5, 2], [3, 5, 2], [3, 5, 2]]
```

Optimal

Page address stream

2 3 2 1 5 2 4 5 3 2 5 2

Optimal



Optimal

```
mx = -1
for k in range(frame):
    mx = max(mx, allocation[i-1][k])
mn = mx
if page_add[i] > page_add[i-1]:
    allocation[i][allocation[i].index(page_add[i-1])] = page_add[i]
else:
    for j in range(frame):
        if allocation[i-1][j] > page_add[i-1]:
            mn = min(mn, allocation[i-1][j])

    allocation[i][allocation[i].index(mn)] = page_add[i]
page_fault = page_fault + 1
```

Optimal

```
Select the method (1 , 2, 3) :

1. FIFO
2. LRU
3. OPT
3
2 3 2 1 5 2 4 5 3 2 5 2
3
--------------------------------------------------------------------------
page fault : 3
[[2, 0, 0], [2, 3, 0], [2, 3, 0], [2, 3, 1], [2, 3, 5], [2, 3, 5], [4, 3, 5], [4, 3, 5], [4, 3, 5], [2, 3, 5], [2, 3, 5], [2, 3, 5]]
```