Palestine Polytechnic University

COLLEGE OF INFORMATION TECHNOLOGY AND COMPUTER ENGINEERING

OPERATING SYSTEMS 7505

FINAL PROJECT

Instructors: Dr.Radwan Tahboub

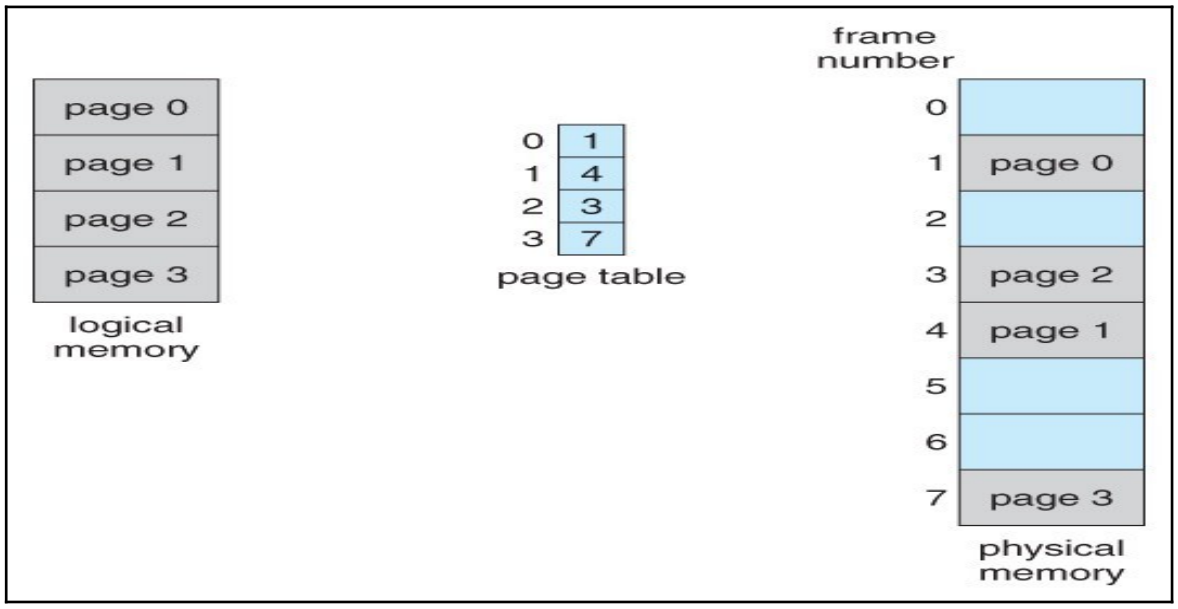
Student : Raghad abu samor

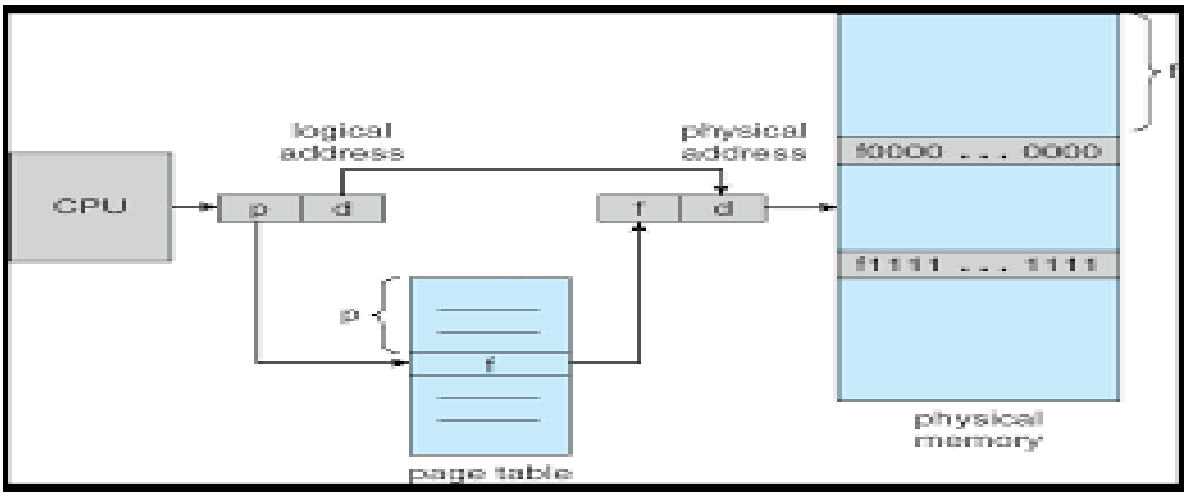
StudentID:217099

Part #2 :Simulate a Paging Memory Manager.

Over view of paging

Paging is a fixed size partitioning scheme.
Divide physical memory into fixed-sized blocks called frames (size is power . of 2)
Divide logical memory into blocks of same size called pages.
keep track of all free frames.
To run a program of size n pages, need to find n free frames and load . program
Set up a page table to translate logical to physical addresse.
internal fragmentation :each process is divided into parts where size of each part is
same as page size.
The size of the last part may be less than the page size.





The simulator :

paging memory and mapping for N processes.

Firstly we defined physical memory as an array, we use some parameters from file such as memory size, page size and process size to calculate the number of free frames in memory which is memory size/page size after finding the number of frames in memory we use vector (v) to hold and track the free frames we initially assign to it a random number in specific range according to the number of frames in memory we do this by using the random function and we use array (arr)and initialize it to zero to prevent redundancy in random numbers generating by assign 1 to arr[random we generated] if the random doesn't use yet. so when we generate random we check if the random was used or not by the condition which checks if(arr[r]==1), if the condition comes true this means we used this random so we should generate anther random, after that, we defined a vector size of it is number of frames in memory, to be a sample of memory to show the mapping and free frames and we initialize it with "Free Frame

Then we sort processes according to arrival time, after that we find the number of pages for each process which is process size/page size then we compare it with the size of vector v which hold the free frames, so if the number of pages is less than or equal the free frames we start building a table to this process so this process will have a table, each page of this process will have a random frame in the memory, we take this random from the vector (v) which contains random numbers represent free frame, after that, we pop this random from v to maintain the number of free frames valid, and we repeat this step until each page in the process take a frame, we assume the

content of each page is "page # for p #" and mapping it to its frame in physical memory by calculating the base which represents the first address of the element of the frame's page in physical memory, also, we mapping page to memory sample instead of showing you all elements of the physical memory which are very large, by erasing the specific frame's of the page because the content of it initially was "Free Frames" then insert in the correct position which is the frame number and fill it with page # for p #", on the other hand if the number of pages for" the process larger than free frames so this process will have not table and we can't mapped it to memory

After that the user will enter a logical address for a specific process, firstly we check the validation of logical address which user enters, by passing it to this condition if logical address < (number of pages of the process *page size), then we find the page number and the offset for the logical address by this equations:

page_no = logical address /page size

offset=logical address % page size

after that we find the physical address by this equation:

physical address = (frame of page_no)*page size + offs

note* we got the frame of the page_no from the table of the / .process

then we print frame number for the page number, offset, and physical addresses

TEST FILE 1

5 // NUMBER OF PROCESS

4096

512

10

1

0 3 10 8192

1 0 12 2048

2 1 3 512

3 5 21 4096

4 9 7 1024

THE OUTPUT

```
pagging memory


table for process  1
|page |0|  frame |0|
|page |1|  frame |5|
|page |2|  frame |4|
|page |3|  frame |2|

table for process  2
|page |0|  frame |3|

process 0 need # pages needs > #of free frames so we cant put it in memory

process 3 need # pages needs > #of free frames so we cant put it in memory

table for process  4
|page |0|  frame |1|
|page |1|  frame |6|
```

NOTE THAT: PROCESS 0 AND PROCESS 3 SIZE (8192 , 4096) IS LARGER THAN MEMORY SIZE (4096)

```
memory mapping of the physical memory

0      -------------------
      |   page0 for p1    |
512   -------------------
      |   page0 for p4    |
1024   -------------------
      |   page3 for p1    |
1536   -------------------
      |   page0 for p2    |
2048   -------------------
      |   page2 for p1    |
2560   -------------------
      |   page1 for p1    |
3072   -------------------
      |   page1 for p4    |
3584   -------------------
      |   free frame      |
4096   -------------------
```

FOR PROCESS 0

```
4096   -------------------
enter process id you want to enter logical address for it  0
this process has not mapped yet
raghad@raghad-HP-250-G7-Notebook-PC:~/Desktop/OS/OS project PART 1$
```

Filter... Output Search and Replace Current Project Terminal Panel

FOR PROCESS 1

```
4096   -------------------
enter process id you want to enter logical address for it  1
enter logical adress for it 1050
(2,26)  -->   (4,26)
physical address  =2074
raghad@raghad-HP-250-G7-Notebook-PC:~/Desktop/OS/OS project PART 1$
```

Output Search and Replace Current Project Terminal Panel

10:32 AM 5/4/22

FOR PROCESS 2

```
4096   -------------------
enter process id you want to enter logical address for it  2
enter logical adress for it 100
(0,100)  -->   (3,100)
physical address  =1636
raghad@raghad-HP-250-G7-Notebook-PC:~/Desktop/OS/OS project PART 1$
```

Filter... Output Search and Replace Current Project Terminal Panel

FOR PROCESS 3

```
4096   -------------------
enter process id you want to enter logical address for it  3
this process has not mapped yet
raghad@raghad-HP-250-G7-Notebook-PC:~/Desktop/OS/OS project PART 1$
```

Output Search and Replace Current Project Terminal Panel

10:35 AM 5/4/22

FOR PROCESS 4

```
enter process id you want to enter logical address for it  4
enter logical adress for it 100
(0,100)  -->   (1,100)
physical address  =612
raghad@raghad-HP-250-G7-Notebook-PC:~/Desktop/OS/OS project PART 1$
```

Filter... Output Search and Replace Current Project Terminal Panel

10:36 AM 5/4/22

THE END

THANK YOU