Assignment 11

Operating System Lab (CS341)

Department of CSE, IIT Patna

Date:- 11-April-2021

Instructions:

1. All the assignments should be completed and uploaded by 13th April 2021, 10 AM.
2. Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
3. Proper indentation and appropriate comments (if necessary) are mandatory. [2+2 marks]
4. You should zip all the required files and name the zip file as roll_no.zip, eg. 1701cs11.zip.
5. Each question will be evaluated on 10 test cases and marks will be given accordingly.
6. Upload your assignment (the zip file) in the following link:

https://www.dropbox.com/request/LA9LcbohAzd1cr7uEes9

Problem :

Assume that the main memory (physical memory) size is 2^32 bits (address range 0 to $2^{\wedge 32}$-1) and frame size is $2^{\wedge 17}$ bits. Design a MMU. It should have following methods:

1. add_process(process_id, req_memory) (time complexity <= O(n))

: process_id (int) -> process id

: req_memory (int) -> required memory in bits

1.1 Use paging to allocate memory

1.2 For each page allocate the first available frame

2. get_address(virtual_address) (time complexity must be O(1))

:virtual_address (32 bit int):

[5-bits:process_id,10-bits:page_offset,17-bits:frame_offset]

2.1 Return the bit address for main memory

Use the above designed system to answer the queries for below inputs.

Input/Output:

First line contains an integer K.

Next line has 2*K integers specifying K frame ranges (inclusive) which cannot be allocated to

any process.

Next line contains integer Q specifying the number of queries.

Next Q lines contain queries.
There are two types of queries:

1. a process_id req_memory

add a process with process id 'process_id' and physical memory of 'req_memory' bits.

2. d virtual_address

given virtual address 'virtual_address' print corresponding physical memory address.

You can assume that all the inputs provided will be in correct format and ranges. For each query of type ‘a’ process id will be different. For each query of type ‘d’ virtual address will be valid.

Sample:

Input:

```
3
0 10 15 20 30 40
3
a 1 64
d 134217735
d 134217748
```

Output:

```
24
48
```

Explanation:

Frame ranges (0, 10), (15, 20), (30, 40) cannot be allocated. Frame ranges are inclusive.

a 1 64

add process with id ‘1’ and physical memory of 64 bits.

d 134217735 `(00001 0000000000 00000000000000111 )` (1, 0, 7) -> address of 7th bit of frame corresponding to 0th page for process with id ‘1’

d 134217748 `(00001 0000000000 00000000000010100 )` (1, 0, 20) -> address of 20th bit of frame corresponding to 0th page for process with id ‘1’