B.Sc. (Hons.) Computer Science
Internal Assignment - Theory
Operating Systems
ANSHUL VERMA (19/78065), Semester - 3

Sol: 1) No. of pages in logical address space = 64 $= 2^6$.

(a.) Bits required for addressing of 1024-word page = 10
[as $1024 = 2^{10}$]

∴ Bits required in logical address
$= 6 + 10 = 16$ bits

(b.) No. of frames $= 32 = 2^5$

∴ 5 bits are required to address frames.

Size of ~~word~~ page $= 1024 = 2^{10}$ words (bytes)

∴ 10 bits are required to address the pages.

∴ Bits required in Physical address
$= 5 + 10 = 15$ bits

Sol: 2) (a) CPU utilization 13% ; disk utilization 97%.

Disk utilization is too high at 97% but CPU utilization is too low at 13%. This will result in thrashing.

Processes, here, are being executed are spending much time on disk in attempting to be paged.

→ The degree of multiprogramming cannot be increased because process here are needed to be suspended to increase CPU utilization.

→ As processes are spending most of their time for paging, the paging is clearly not helping.

(b) CPU utilization 87% ; disk utilization 3%.

→ This means CPU is very busy. It ~~was~~ is limiting the proper disk utilisation and will be bottleneck eventually.

→ The degree of multiprogramming, thus, cannot be increased as increasing it would inevitably cause thrashing.

→ As the CPU is being kept busy → number of processes increases → Thus, paging is helping.

(c) CPU utilization 13% ; disk utilization 3%.

→ CPU and disk space, both utilization is low.

→ ~~The degree of programming can and sh~~
~~CPU is not working~~

→ The degree of multiprogramming can and should be increased to increase CPU utilization as the CPU is not working with enough capacity that it can.

→ As the number of processes are less → paging is not having any reasonable effect

Sol:3) The effect is that two pages refer to same physical page/address.

The advantage of this scheme is not having to make any copies of same data in main memory and this is acheived by sharing the data.

→ Lets say we have to copy a large amount of code or data from one place in the memory to other. Instead of actually duplicating the same data if we use the above scheme, and just create another page table entry to location of data, time needed would be much less as we are not actually allocating any new memory.

→ However, if in this scheme, some byte is updated from one ~~entry~~ page entry, it will be ~~reflected~~ reflected on other, which means it will change the data in physical memory and since, its address is shared, the changes will be reflected on other pages.

Sol:4) ① An FTP server, from which we can download multiple files as well as upload ~~same~~ a file simultaneously.

This can be easily acheived in multithreading scheme by creating threads which handles each task of download/upload seperately.

Also, In single threading scheme, if one ~~download~~ task fails, other tasks following it won't be executed but this is not the case with multithreading scheme.

(2) A parallelized application such as matrix multiplication where different parts of matrix may be worked on in parallel.

(3) A video game, in which we are controlling the player with keyboard input as well as mouse and ~~shooting~~ performing another task (like shooting).

All this can be done parallely in multithreading scheme.