

Operating Systems (CSL 3030)

Readme file

Lab Assignment	05
Roll No.	B20CS018
Name	Harshita Gupta

How to run:

1. Open the directory where the code is located.
2. Compile and run the execution of the C file using the commands -

```
gcc demandpaging -o dpaging.exe  
./dpaging.exe
```

Input contains 4 space-separated integers k m f s , with the number of processes as k , the maximum number of pages required per process as m , the total number of frames in main memory as s and size of the TLB as f .

Assumptions :

1. $k > m > f > s$ and $s > 2$.
2. At Least 1 free frame is common to all processes in the free frame list for a case when random allocation may allocate 0 frames.
3. Local replacement policy is used. Replacement hence will be local other than for common free frames.

Descriptions:

Demand paging simulation is implemented in the code for memory management. The frames were allocated to the processes according to demand and a portion of frames were kept free. LRU policy is being used to update the page table as well as the TLB in case of no free frame availability. For the initial allocation of frames to each process, proportional allocation is used.

Observations:

1. Number of TLB hits increases with the increase in TLB size. However, this change

is reflected upon large-sized processes only for example 2nd and 3rd processes in the result file.

2. Page faults decrease as the number of frames is increased. It should be noted that for this observation to be significantly seen, reference strings should be sufficiently long. For smaller reference strings, a few starting references will always have page faults.
3. If the process is referencing similar or nearby memory addresses consecutively, then the TLB hits are more common. Since TLB size is not very large, thus referencing memory addresses in the spatial locality of the process leads to lesser TLB misses and in turn lesser page faults.