README

-Mayank Raj(B19CSE053)

Implementation details:

- Everything including TLB, free frame list are implemented and page fault handling is used using the LRU algorithm.
- The maximum size of the TLB is taken as input.
- Demand paging correctly implemented(Verified manually on different inputs).
- Results stored in results.txt

How to run:

- Open a terminal and write
- g++ DemandPaging1.cpp;./a
- Enter the inputs as specified=>
  k=No. of processes
  M=maximum no. of pages per process
  F=Frames in main memory
  S=TLB maximum size
- Check result.txt for results

Results:

- For each process, a reference string is generated, and first, its searched in TLB and if a TLB miss occurs it searches in page Table, and if a page fault occurs it brings that page in main memory using the LRU algorithm.

- Searching in TLB takes lesser time, so they can be used as cache lookup tables. For a new process, the previous content of TLB is flushed for new data to come in.
- Sequential details of TLB hit/miss and page hit/miss and page faults for each process.
- No. Page faults are mentioned.

Limitations or bugs:

- If the maximum size of TLB is too low then there won't be much improvement in time as TLB miss will be high and eventually it has to look into the page table.

Sample input for results.txt: 4 8 5 2

You can enter your input as well.