

OS Project2 README 2019/2020

Thodoris Maximilianos Chytis sdi1700197

Files:

- source : main.c, vmemfuncts.c
- headers: types.h, vmemfuncts.h
- input: bzip.trace, gcc.trace
- output: entries.txt

This project successfully simulates the Virtual Memory of an Operating System by using an Inverted Paging Table (IPT) and the famous page replacement algorithms, Least Recently Used (LRU) and Working Set (WS). In order to make the simulation believable the program reads traces from two different trace files bzip.trace and gcc.trace as if there were two different processes. As parameters the program is given the wanted Page Replacement algorithm, the number of frames the IPT should consists of, the number of addresses each process reads before switching to the other process, the maximum number of addresses that shall be read by both processes and if the wanted page replacement algorithm is WS the size of the needed window, in that order.

The IPT is allocated by using a pointer pointing to the beginning of f_num pointers pointing to a struct page.

This struct consists of the process id of the process having asked for a page, the Reference bit, first 5 character of the trace being the page address. And the timestamp of this entry in the paging table.

```
struct page{  
    int pid;  
    int valbit;  
    char* page;  
    int timestamp;  
};
```

Then the program starts reading q traces using each process from the two trace files, until either the max number of traces to be read is

reached or both files reach EOF.

The function called by each Process to ask for pages is the IPTManager which checks for the address in linear fashion inside the IPT before updating it if needed using the correct page replacement algorithm

LRU: The moment the IPT is full and we get a PAGEFAULT, the oldest address in the IPT is replaced by the one asked for by a process.

WSUpdate: The moment the IPT is full and we get a PAGEFAULT, all addresses from the same process that have been in the IPT with “count - IPT → timestamp > window” (longer than the WS window allows) get removed and the page being asked for is inserted into the IPT at some now free index.

At the end the program gives statistics about the number of reads, the number of writes, the number of PAGEFAULTS, the addresses that

were examined from the trace files (in a file called entries.txt) , and the frame number of the IPT.

All allocate memory is deallocated, and the files are closed.

Compilation:

1)**LRU**

make

./VMEMSimluator LRU 3 4 8

make clean

make

./VMEMSimluator LRU 7 10 51

make clean

make

./VMEMSimulator LRU 40 5 236

make clean

2)WS

make

./VMEMSimluator WS 3 4 8 2

make clean

make

./VMEMSimluator WS 7 10 51 4

make clean

make

./VMEMSimulator WS 40 5 236 21

make clean