

COMP3300 OS Fundamentals Project

Guo Ziheng

April 5, 2021

I confirm that I will keep the content of this assignment confidential. I confirm that I have not received any unauthorized assistance in preparing for or writing this assignment. I acknowledge that a mark of 0 may be assigned for copied work. Student ID: 105148920, name: Guo Ziheng.

1 Project implementation

The author has paid the most effort to make the code as readable as possible, including naming the variables and functions with readability, explaining function usage, extra comments, etc. Here we will give a quick explanation on this project. Please go to the code to confirm more details.

We will use `char[]` to represent physical memory. Hence we apply some memory space using system call `malloc()`. As required, the memory size will be `MAX_PAGE*PAGE_SIZE`, which is 256×256 . The index of this array will be the physical address.

The major part of the program is simple: read every required logical address, convert them to physical address, then print what is inside the physical address. `convertAddress()` is built to grantee that the required logical address is already in physical memory. In another word, it will do the loading part.

`convertAddress()` will take the high 8 digit as the page number, and call `findPage()` to find the frame of the required page. Once the frame is found, simply change the upper 8 digit. Some trick of bit operation is used here.

`findPage()` will simply check the page table one by one. If found the required page, return the frame(line 69-70). If the page tale is empty(line 64-65) or come to the last entry without finding it(line 72-73), it will call `loadPage()`.

`loadPage()` accept the pointer to `lastEntry` and the required page. It will create a page-frame relation entry, attach it to `lastEntry`, set up the new page-frame relation, then load from `BACKING_STORE.bin` to the frame.

An array of `frameOccupancy[]` is used for tracking frame usage. The first empty frame will be selected when loading new page.

2 Result

The code compiles with no error or warning. Compiled on `Windows Subsystem Linux(WSL)`, kernal version `4.4.0-19041-Microsoft`. Compiled program and running result is attached within the submission. As we can see in the result, it has completely same result as the provided `correct.txt`.