Operating Systems – 2: CS3523
Spring 2023
Report- Assignment 6: Paging

Name: Ishaan Jain

Roll Number: CO21BTECH11006

General Instructions

- There are 2 files in the archieve file submitted:
  - xv6.tar.gz
  - Report.pdf

- To compile and run the xv6 os, go to the directory xv6 and run the following commands:
  - $ make
  - $ make qemu

- To run demand paging, write the following command in the xv6 booted terminal:
  - $ demandpaging

Part 1:

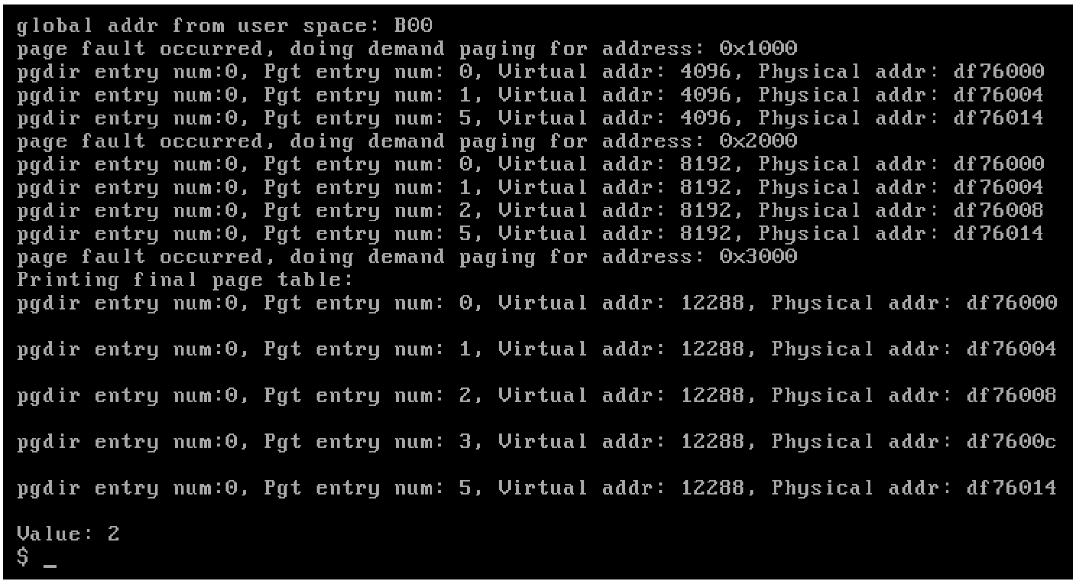
Program Design

- The following changes are made in the exec.c file from line 53-56:
  - Here a page fault is created
    ```
    if ((sz = allocuvm(pgdir, sz, ph.vaddr + ph.filesz)) == 0)
        goto bad;
    sz += ph.memsz -ph.filesz;
    ```
- Created a case for page fault (T_PGFLT) in trap.c
  - The case is taken from allocuvm() which allocates page tables and physical memory
  - This case prints a message of a page fault when it occurs and calls map_page() function which is exactly the same as mappages() to map memory to those pages.
- map_page() is defined in vm.c at the end. It is exactly the same as mappages() in the same file.
- The user level program demandpaging.c is implemented. The code for that was already given to us.
- The declaration of map_page() is made in the def.h file.
- The _demandpaging command is added to UPROGS in Makefile code in line 186.

- The previous sys_mypgtPrint() in sysproc.c in my last submission was altered a bit to get the required print format. The virtual address is taken from rcr2().

Working of the code:

- The exec.c doesn’t allocate memory to dynamic variables when calling allocuvm.
- This will cause the program to create page faults.
- The case T_PGFLT in trap.c is under execution now.
- It gets the address where the pagefault has occured and then then assigns memory.
- It then maps the pages to physical memory and virtual memory.



```
global addr from user space: B00
page fault occurred, doing demand paging for address: 0x1000
pgdir entry num:0, Pgt entry num: 0, Virtual addr: 4096, Physical addr: df76000
pgdir entry num:0, Pgt entry num: 1, Virtual addr: 4096, Physical addr: df76004
pgdir entry num:0, Pgt entry num: 5, Virtual addr: 4096, Physical addr: df76014
page fault occurred, doing demand paging for address: 0x2000
pgdir entry num:0, Pgt entry num: 0, Virtual addr: 8192, Physical addr: df76000
pgdir entry num:0, Pgt entry num: 1, Virtual addr: 8192, Physical addr: df76004
pgdir entry num:0, Pgt entry num: 2, Virtual addr: 8192, Physical addr: df76008
pgdir entry num:0, Pgt entry num: 5, Virtual addr: 8192, Physical addr: df76014
page fault occurred, doing demand paging for address: 0x3000
Printing final page table:
pgdir entry num:0, Pgt entry num: 0, Virtual addr: 12288, Physical addr: df76000

pgdir entry num:0, Pgt entry num: 1, Virtual addr: 12288, Physical addr: df76004

pgdir entry num:0, Pgt entry num: 2, Virtual addr: 12288, Physical addr: df76008

pgdir entry num:0, Pgt entry num: 3, Virtual addr: 12288, Physical addr: df7600c

pgdir entry num:0, Pgt entry num: 5, Virtual addr: 12288, Physical addr: df76014

Value: 2
$ _
```

The above screenshot is a sample output after passing demandpaging command in the qemu terminal.