Programming Assignment 3 Part B

Binghan Geng A20482350
bgeng1@hawk.iit.edu

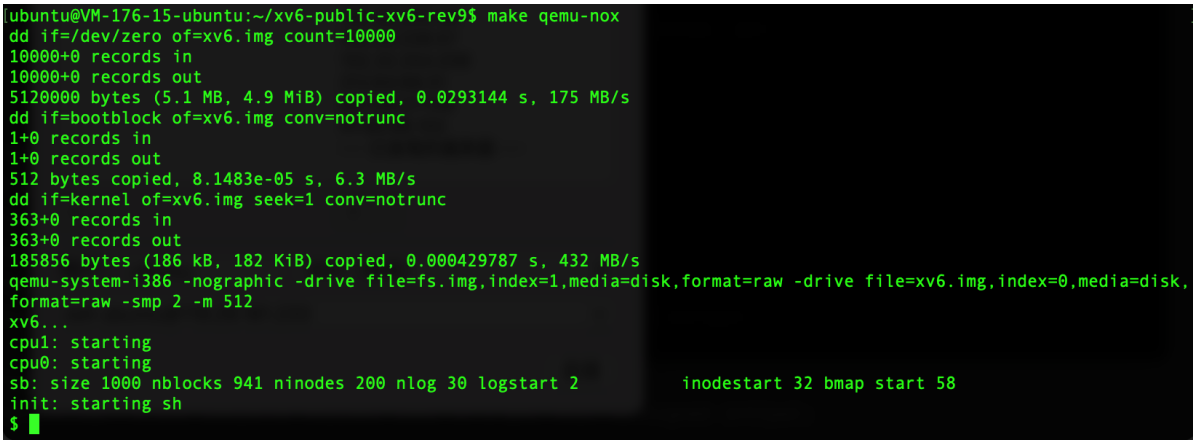
Yu Li A20496405
yli385@hawk.iit.edu

Introduction

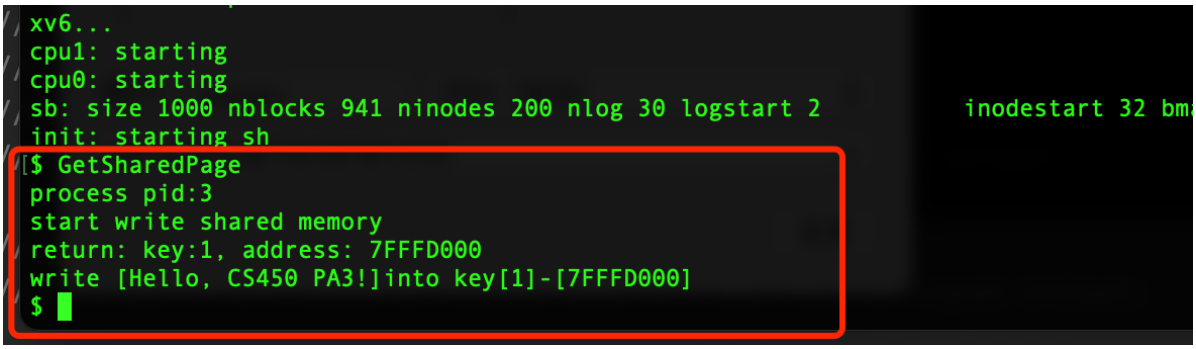
This part is to learn how to implement a pair of xv6 system calls: GetSharedPage() and FreeSharedPage() that will allow two programs (two processes) to share pages.

Build & Run Instructions

compile: make clean
make qemu-nox



execute systemcall: GetSharedPage and FreeSharedPage



```
process pid:3
start write shared memory
return: key:1, address: 7FFFD000
write [Hello,CS450 PA3!]into key[1]-[7FFFD000]
$ FreeSharedPage
process pid:4
start read shared memory
return: key:1, address: 7FFFD000
start release shared memory
FreeSharedPage: key is 1
FreeSharedPage: refcount is 2
FreeSharedPage: page_nums is 3
Free the user space memory.
free shared memory: key[1]-[7FFFD000]
```

execute Test Case: my_shell 3 3

```
GetSharedPage  2 18 12912
GetShmByParam  2 19 13140
FreeSharedPage 2 20 13072
FreeShmByParam 2 21 13380
my_shell       2 22 24020
console        3 23 0
$ my_shell 3 3
process pid:4
param: key:3, num_pages: dx
begin: share a memory page, key:3
return: key:3, address: 7FFFD000
write [Hello,XV6!]into key[3]-[7FFFD000]
```

GetShmByParam 3 3

```
CS450$ GetShmByParam 3 3
process pid:5
start write shared memory
param: key:3, num_pages: dx
return: key:3, address: 7FFFD000
write [Hello,CS450 PA3!]into key[3]-[7FFFD000]
```

FreeShmByParam 3 3

```
CS450$ FreeShmByParam 3 3
process pid:6
start read shared memory
param: key:3, num_pages: dx
GetSharedPage Return: key:3, address: 7FFFD000
read [Hello,CS450 PA3!] from key[3]-[7FFFD000]
start release shared memory
FreeSharedPage: key is 3
FreeSharedPage: refcount is 3
FreeSharedPage: page_nums is 3
Free the user space memory.
free shared memory: key[3]-[7FFFD000]
```