Gloria Nduka
April 2021
Project 4

The files I edited in the xv6-public-master folder are bolded below.

syscall.c
Included extern int sys_procdump(void); line 106
In *syscall[] library/array [SYS_procdump] sys_procdump, LINE 130. Added procdump to syscall table.

syscall.h
I assigned procdump a unique number.
Line 23
#define SYS_procdump 22
This file contains symbolic definitions of system call numbers. Define a unique number for the system call.

sysproc.c
I implemented int sys_procdump(void) in lines 93-97. I returned/called procdump();

user.h
I included the system call function prototypes. I added int procdump(void) on line 26.

usys.S
I included SYSCALL(procdump) on lines 32.

Makefile
I added the cow test in the UPROGS variable and the EXTRA variable so it can compile with all the other files when running make.

mmu.h
I created a shared page tag on line 98 and initialized it to be 0x200.

proc.c
I added to the procdump function in lines 533-550. I implemented the enhanced process details viewer. I displayed the virtual page number, the physical page number, and whether it is writable or not writable. I iterate through the process then the pages then check if it is user and present then mask and shift the first 20 bits then return yes if it is writable then return no if not.

testcow.c

I included the provided testcow.c to the folder. This file basically calls procdump and tests COW implementation. In line 11, malloc is called to allocate the defined size of the space variable. The fork function creates a new child process that's identical to its parent process and then calls procdump() for that child process before and after values have been changed.