file\_op.c:

stdin\_read:

return data is not correct ( I simply return nbyte at the beginning) which cause the the user program to write out of the boundary of the buffer. Sometime change the stored fd to 0 and cause problem.

stdout\_write:

I first directly used the user buffer and in order to use printf function, add a ‘\0’ to the end of the buffer. It turns out that I write out of the boundary of user buffer and changed the fd value to 0

Later use a “naive kmalloc function” to create temp buffer. It works fine before cp5 but create unprotected condition when multitasking was introducing into our kernel.

Filesys

Read\_data:

Too many bugs for me to remember. Including return length is not correct, not reading enough code

Idt/idt\_init:

Error code cannot be read correctly. At the beginning I thought I have to push a 0 for the error code field for every isr. It turns I only need to do this for those that doesn’t have error code

Kmalloc

Kmalloc: doesn’t take care of align which cause leak of memory

r\_kmalloc/r\_kfree: miscalculate the index

lbuffer:

Variable defined in this file doesn’t work with other files (mistake when using keyword extern)

Lib:

Miscalculate the position( for printing to screen)

Misscalutate the shift when reach the last line of screen

display\_video: sometime the background program print to the current terminal. It turn out that this function doesn’t have an algorithm to protect to video memory

paging/paging\_init:

doesn’t know that the pde should align to 4096 which cause triple fault

pid:

initial\_all\_pid: miscalculate the esp value(typo)

sched:

scheduler:some bugs happened here, probably with cr3 value but I cannot remember the detail

syscall.c:

trap\_halt: page that is freed by this function is wrong, causing triple faults

trap\_execute: too many that I can’t remember

using mypcb as curpcb

using wrong cs, ds value

syscall\_Wrap:

pushed the wrong data for Iret

the tracking of stack is not correct so sometime cannot iret back to the correct position

some time switch back to the kernel pde which causes page fault for reading user data