Lab10-Session2-Project3: Dynamic Kernel Linker Report

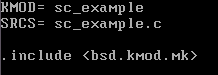
Yukui Ye

SUID:439644268

**Task1:**

a:Souce code which represents the first system call: sc\_example.c

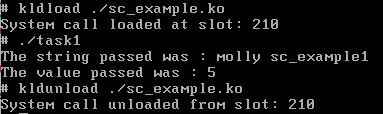
b:Makefile:



c: User level program: task1.c; do gcc –o task1 task1.c

d: compile these code, and load the module into the kernel, after load it, execute user level program, then unload it.

e: Result:

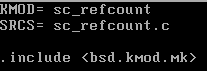


**Task2:**

a:Souce code which represents the second system call: sc\_refcount.c

The code calculate the number of vm\_object in current system has certain reference count, and it prints out all the number based on different reference count.

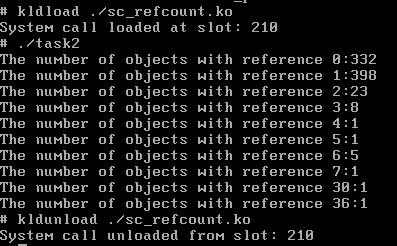
b:Makefile:



c: User level program: task2.c ; do gcc –o task2 task2.c

d: compile these code, and load the module into the kernel, after load it, execute user level program, then unload it.

e: Result:



Analysis:

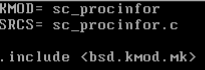
From the above screen shot, we can tell that in current system, there are 10 kinds of processes has been referenced based on different reference count. For example, there are 23 objects which reference count is 2.

**Task3:**

a:Souce code which represents the third system call: sc\_procinfor.c

The code implement a new system call which takes an integer(process ID) as input and collects the some information about the particular process.

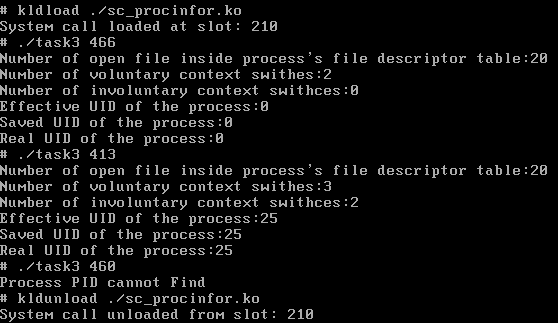
b:Makefile:



c: User level program: task3.c; do gcc –o task3 task3.c

d: compile these code, and load the module into the kernel, after load it, execute user level program, then unload it.

e: Result:



we can tell from the result that process 466 is a root process therefore its effective, save, real UID are all 0; process 413 is not root process, and its effective UID is 25 as well as its saved UID and real UID; Besides from the third example we are trying to look for process with PID 460 but we cannot find it, so it comes out the sentence”Process PID cannot find”. This is implemented in the user level program with an returned integer a from syscall function. Specifically , you can check both the source code sc\_procinfor.c and task3.c