**120090516 YangYin**

**Assignment 1 Report**

**Program1:**

Design

1. Use fork function to fork a child process.
2. Check whether the process is parent process or child process
   1. Pid=0: Child process. Execve the test program and raise SIGHLD signal.
   2. Pid!=0: Parent process. Wait for child process terminates and output how did the child process terminates and what signal was raised in child process.

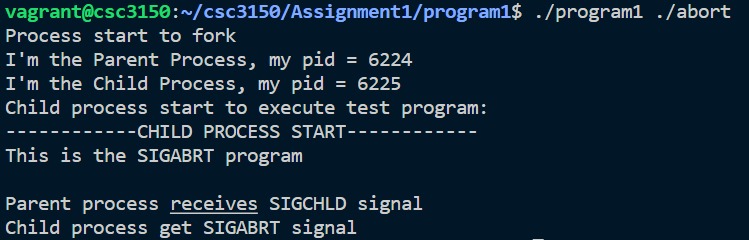
Set Up Environment

I used a VM environment (with Linux Kernel Version 5.10.146, GCC version 5.4.0).

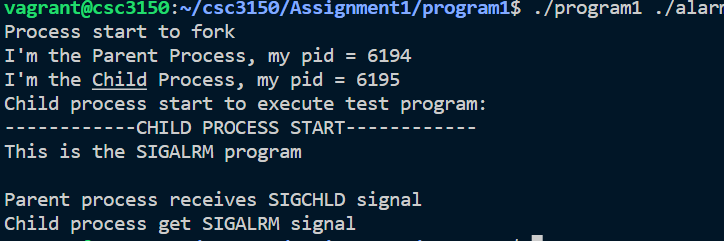
Makefile is used to compiler the program.

Output

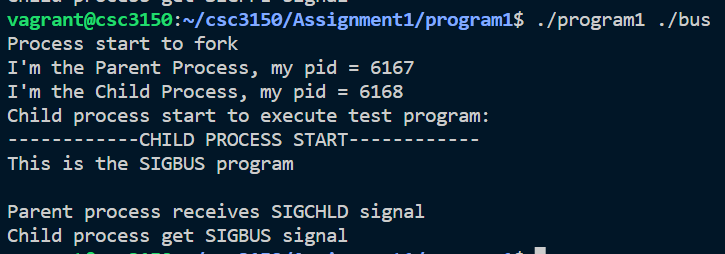
abort



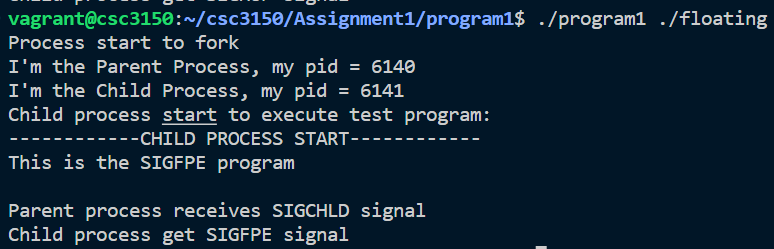
Alarm



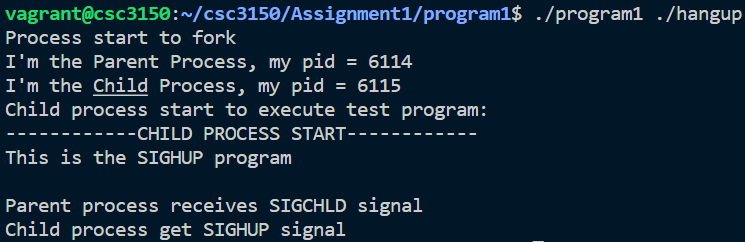
bus



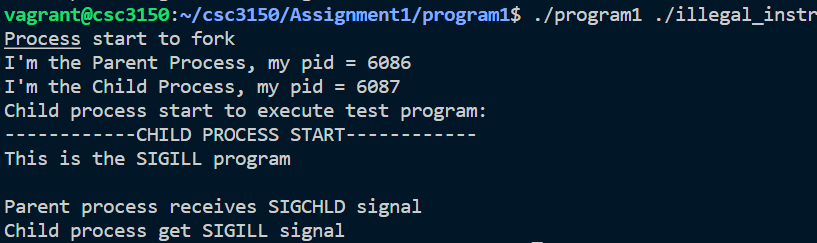
Floating



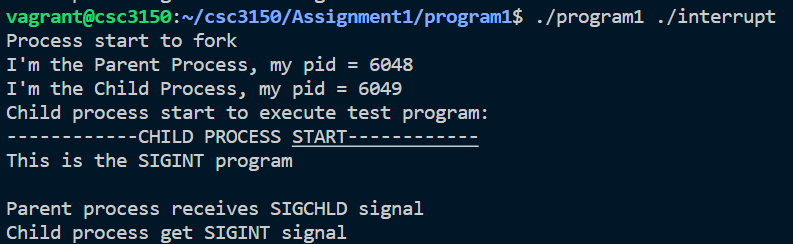
Hangup



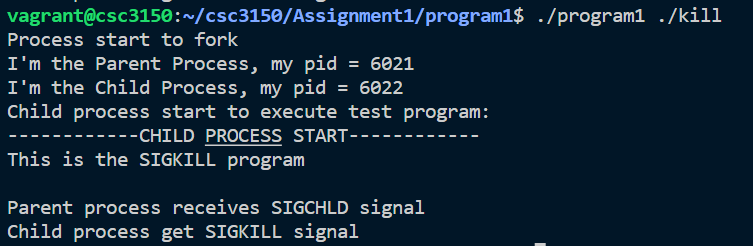
Inllegal\_instr



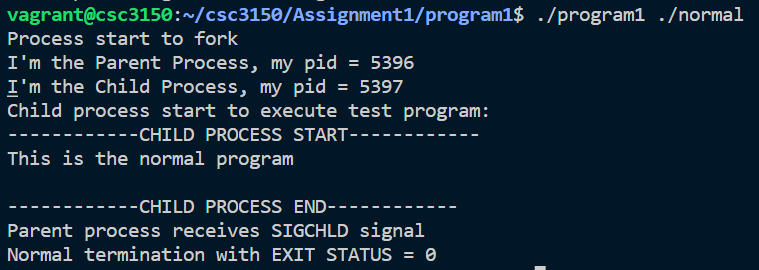
Interrupt



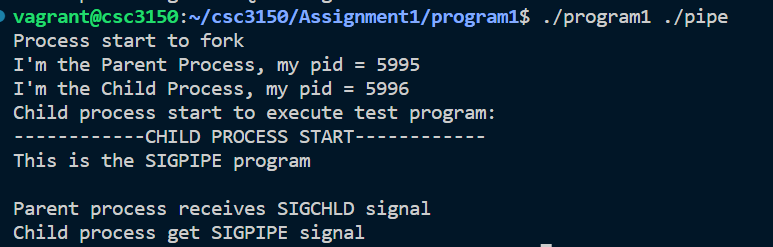
Kill



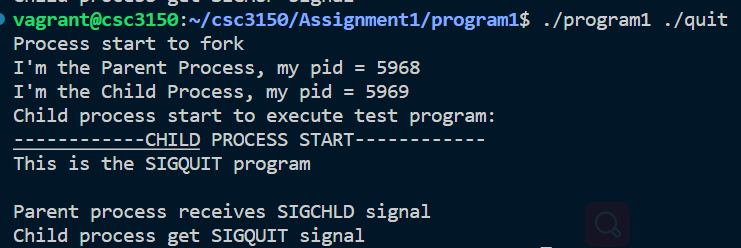
Normal



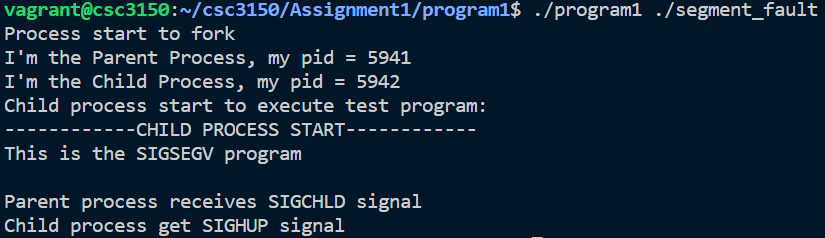
Pipe



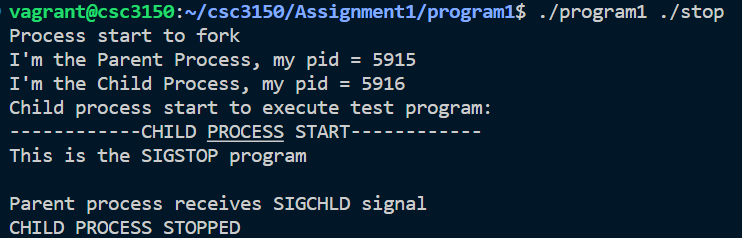
Quit



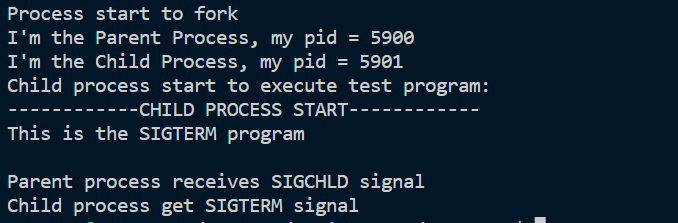
Segment\_fault



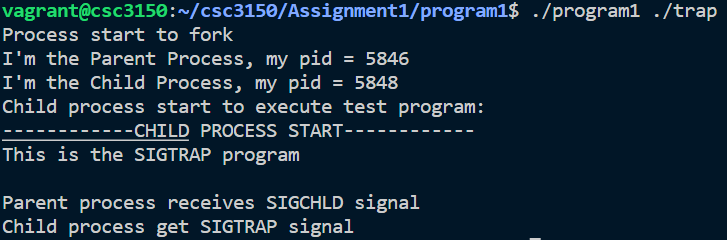
Stop



Terminate



Trap



**Program2:**

Design

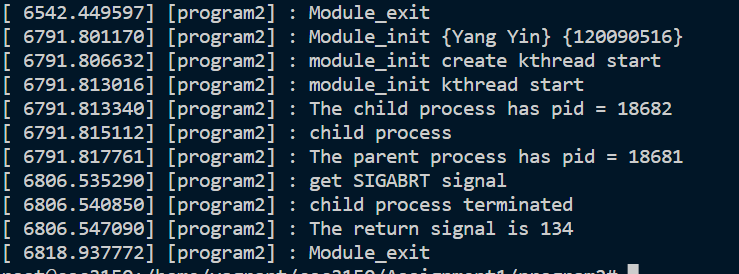
1. Initialize a program2 module and when it is initializing, use kthread\_create function to create a thread run my\_fork function.
2. Within my\_fork, use kernel\_clone to fork a new process which is used to run my\_exec function.
   1. Output the pid of both child and parent process in kernel.
   2. Use parent\_wait function to wait for the child process terminate.
3. Within my\_exce function,
   1. Use getfilename\_kernel to get the test file name.
   2. Use do\_exceve to exceve the test program.
4. Within parent\_wait function, use do\_wait to wait child process terminate. After waiting, output the signal was raised in child process in kernel.

Set Up Environment

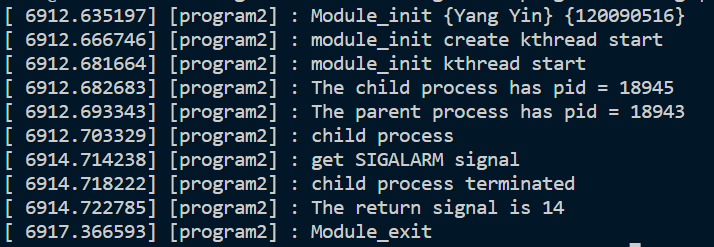
1. Use chmod 777 to edit the kernel code.
2. Export kernel\_clone, getfilename\_kernel, do\_execve, do\_wait function.
3. Build kernel Image and modules, install kernel modules and kernel.
4. Reboot the VM.

Output

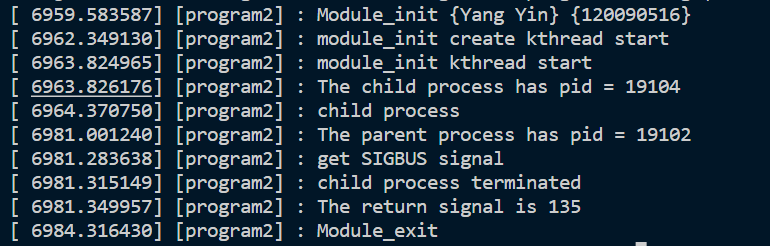
Abort



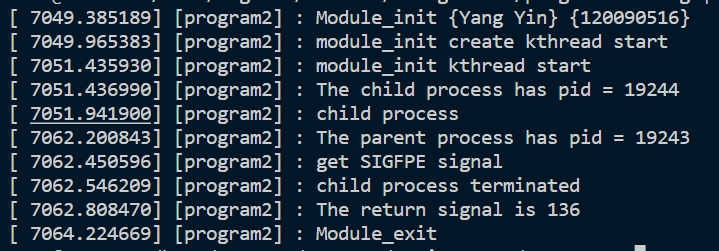
Alarm



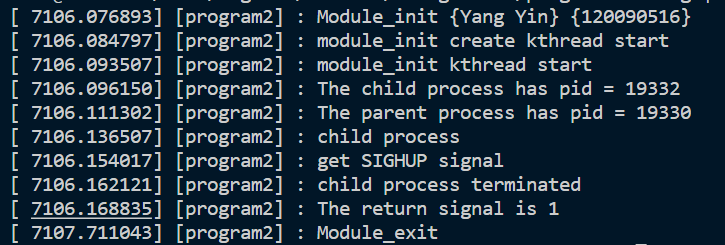
Bus



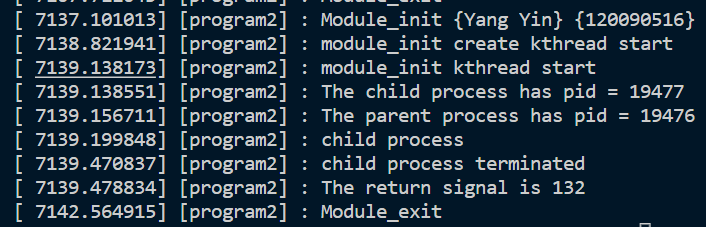
Floating



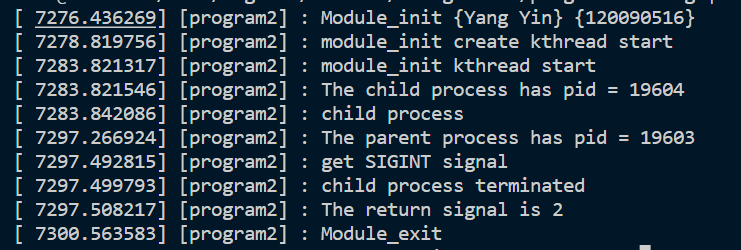
Hangup



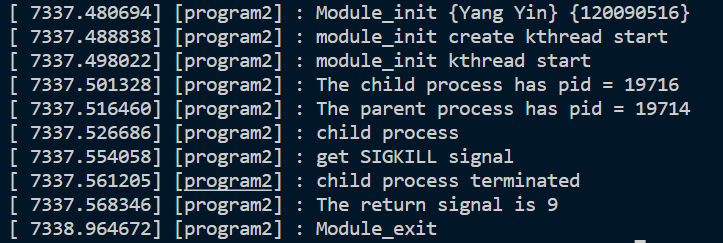
Illegal\_instr



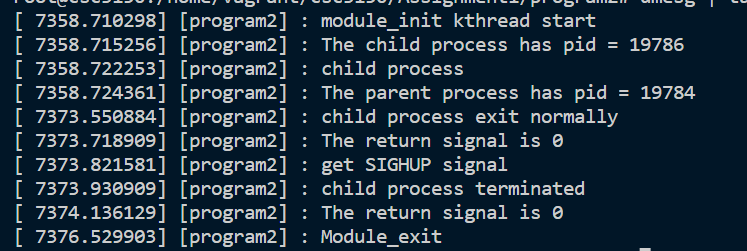
Interrupt



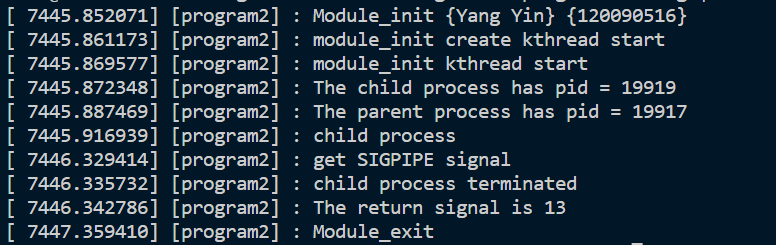
Kill



Normal



Pipe



**NOTES**