CSC3150 Report of Assignment 1

刘翰齐 120090872

How I design the program?

1. The first step is to determine the purpose of the program design.
2. Then design a process to solve the problem.
3. Design the framework according to the process.
4. Complete the code.
5. Try to compile the code and debug.

How to set up the development environment?

My device is an Intel based MacBook.

1. Install the virtual machine (VMware 19).
2. Install Ubuntu operating system (22.04.1, amd64).
3. Install gcc using apt tool.
4. I used Visual Studio Code to edit the code and its plugin for clang formatting.

Screenshot of the program output.

Program 1:

Normal

```
lab24@lab24:~/桌面/source/program1$ ./program1 ./normal
Process start to fork
I'm the parent process, my pid = 24195
I'm the child process, my pid = 24196
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the normal program

------------CHILD PROCESS END------------
Parent process receives SIGCHLD signal
Normal termination with EXIT STATUS = 0
```

Abort

```
lab24@lab24:~/桌面/source/program1$ ./program1 ./abort
Process start to fork
I'm the parent process, my pid = 24204
I'm the child process, my pid = 24205
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGABRT program

Parent process receives SIGCHLD signal
Parent process receives SIGABRT signal
```

Stop

```
lab24@lab24:~/桌面/source/program1$ ./program1 ./stop
Process start to fork
I'm the parent process, my pid = 24207
I'm the child process, my pid = 24208
Child process start to execute test program:
------------CHILD PROCESS START------------
This is the SIGSTOP program

Parent process receives SIGCHLD signal
CHILD PROCESS STOPPED
```

Program 2:

```
lab24@lab24:~/桌面/source/program2$ ./program2
[21138.285211] [program2] : module_init {Hanqi LIU} {120090872}
[21138.285213] [program2] : module_init create kthread start
[21138.285216] [program2] : module_init kthread start
[21138.285218] [program2] : The child process has pid = 5670
[21138.285221] [program2] : This is the parent process, pid = 5664
[21138.285228] [program2] : child process
[21138.285232] [program2] : get SIGTERM signal
[21138.285235] [program2] : child process terminated
[21138.285239] [program2] : The return signal is 15
[21138.285243] [program2] : module_exit
```

What did I learn from the tasks?

1. Modular programs enable a clear logical framework.
2. Programs with step-by-step output are friendly for debugging.