Report for CSC3150 - Assignment 1

Name: Wang Chaoren

Student ID: 122090513

How did you design your program? (4 points)

For Task 1:

I designed this program to create a child process using `fork()` and handle `SIGCHLD` signals to notify the parent when the child's state changes. The child runs an external program with `execl()`, while the parent monitors it using `waitpid()` to check if the child exited normally, was terminated, or stopped by a signal.

To make the output easier to understand, a `map_status()` function that converts signal numbers into readable names.

For Task 2:

I created a kernel thread that forks a child process using `kernel_clone()`. When the module is loaded, the `program2_init()` function starts a thread to run `my_fork()`, which sets up default signal handling (with in code template), creates the child process, and waits for it to finish using `kernel_wait()`.

The child process runs `my_exec()`, where it attempts to execute a test program located at `/tmp/test` using `kernel_execve()`. Print statements were added to display both the parent and child process IDs, helping track their relationship and execution in the kernel logs.

To record different termination scenarios, a `map_status()` function was used like Task 1.

Bonus Task:

I wrote this C program to visualize the process tree in a Linux system by reading from the `/proc` filesystem. I define a structure, `process_node_t`, to store information like the process ID (`pid`), parent process ID (`ppid`), name, and command line. It gathers these details for each process/threads.

After collecting, a tree structure was built in `build_process_tree()` by linking child processes to their respective parents. Depending on the flags, like `sort_by_pid`, the program can sort child processes by name or ID to make the output more organized.

The `print_process_tree()` function then displays the process hierarchy, with options to include PIDs, command line arguments, or use ASCII for a simpler view.

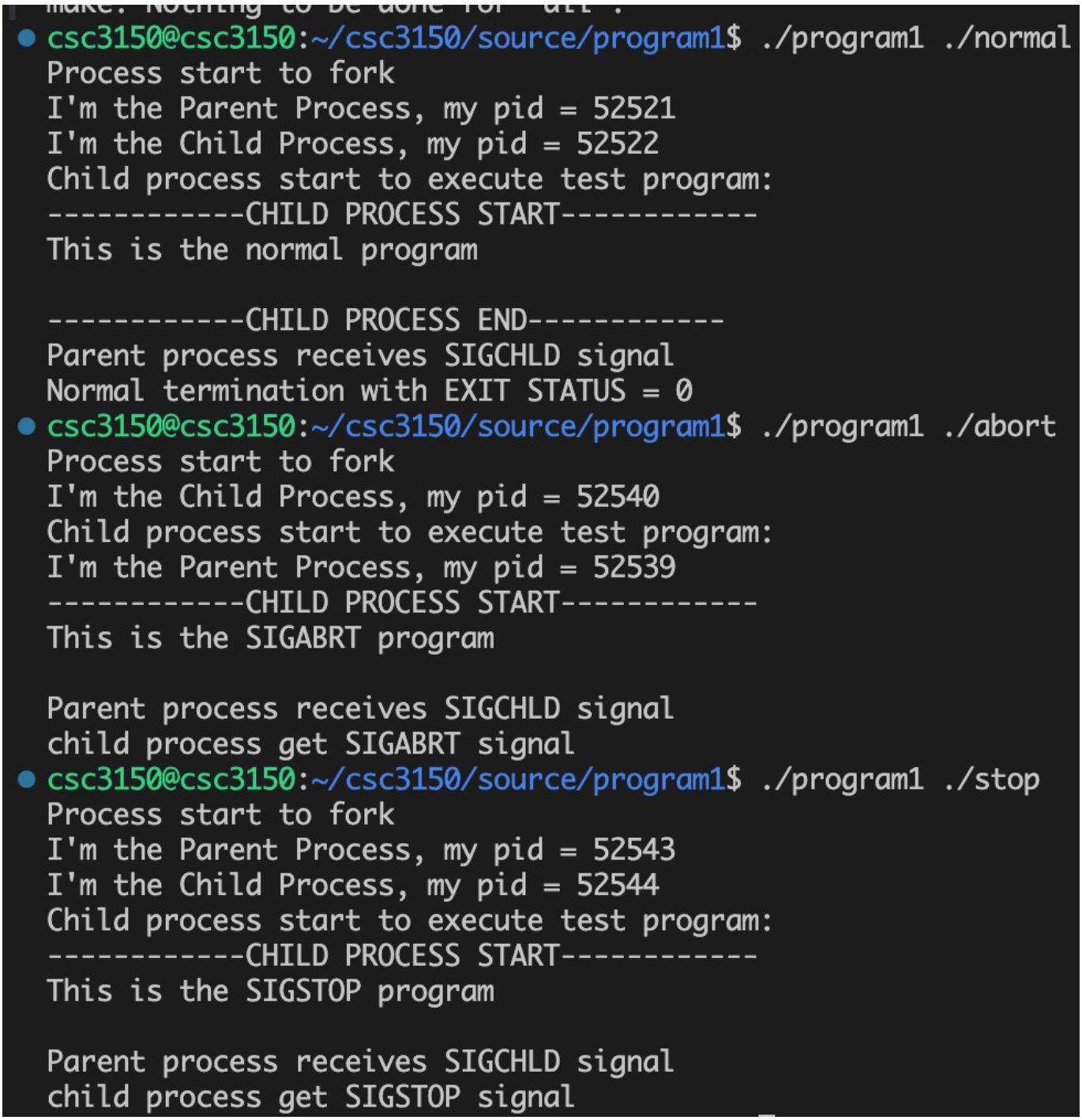
How to set up your development environment, including how to compile kernel? (2 points)

I follow the guide provided by the Tutorial 1 to set up the development environment.

1. Download && Install UTM && Load Image
2. Install the necessary packages
3. `scp` the template code to the VM
4. Downloading kernel source code 5.15.10 && Compile the 5.15.10 kernel according to the tutorial.
5. Failed to build due to disk space issue, then increate the disk space, follow this.
6. Failed to build, certs error, follow this to fix it.
7. Failed to build, due to `pahole` wrong version, follow this to fix it.
8. Successfully build the kernel.
9. Load the kernel into VM, and boot it.
10. Failed to start the system, unknown reason.
11. Try again to build 5.10.1 kernel.
12. Cannot build, due to `pahole` issue, previous fix does not work. Follow this to disable `BTF`.
13. Successfully build the kernel.
14. Load the kernel into VM, and boot it.
15. For better development experience, using `vscode` to connect the SSH.
16. Possible network issue, failed to start vscode, manually downloaded the vscode server and `scp` to the VM.

Screenshot of your program output. (2 points)

Task 1:



Task 2:

Remember to copy the executable file to `/tmp/test` before running the module.

```
cp executable_file /tmp/test
```

```
root@csc3150:/home/csc3150/csc3150/source/program2# make
make -C /lib/modules/5.10.1/build M=/home/csc3150/csc3150/source/program2 modules
make[1]: Entering directory '/home/seed/work/linux-5.10.1'
make[1]: Leaving directory '/home/seed/work/linux-5.10.1'
root@csc3150:/home/csc3150/csc3150/source/program2# insmod program2.ko
root@csc3150:/home/csc3150/csc3150/source/program2# rmmod program2.ko
root@csc3150:/home/csc3150/csc3150/source/program2#  dmesg | tail
[69192.599896] [program2] : Module_init
[69192.600542] [program2] : module_init create kthread start
[69192.601485] [program2] : module_init kthread start
[69192.601697] [program2] : The child process has pid = 54165
[69192.613002] [program2] : This is the parent process, pid = 54163
[69192.622978] [program2] : child process
[69192.646855] [program2] : get SIGTERM signal
[69192.647069] [program2] : child process terminated
[69192.647070] [program2] : The return signal is 15
[69194.569799] [program2] : Module_exit
```

Bonus Task:

Supported 7 arguments:

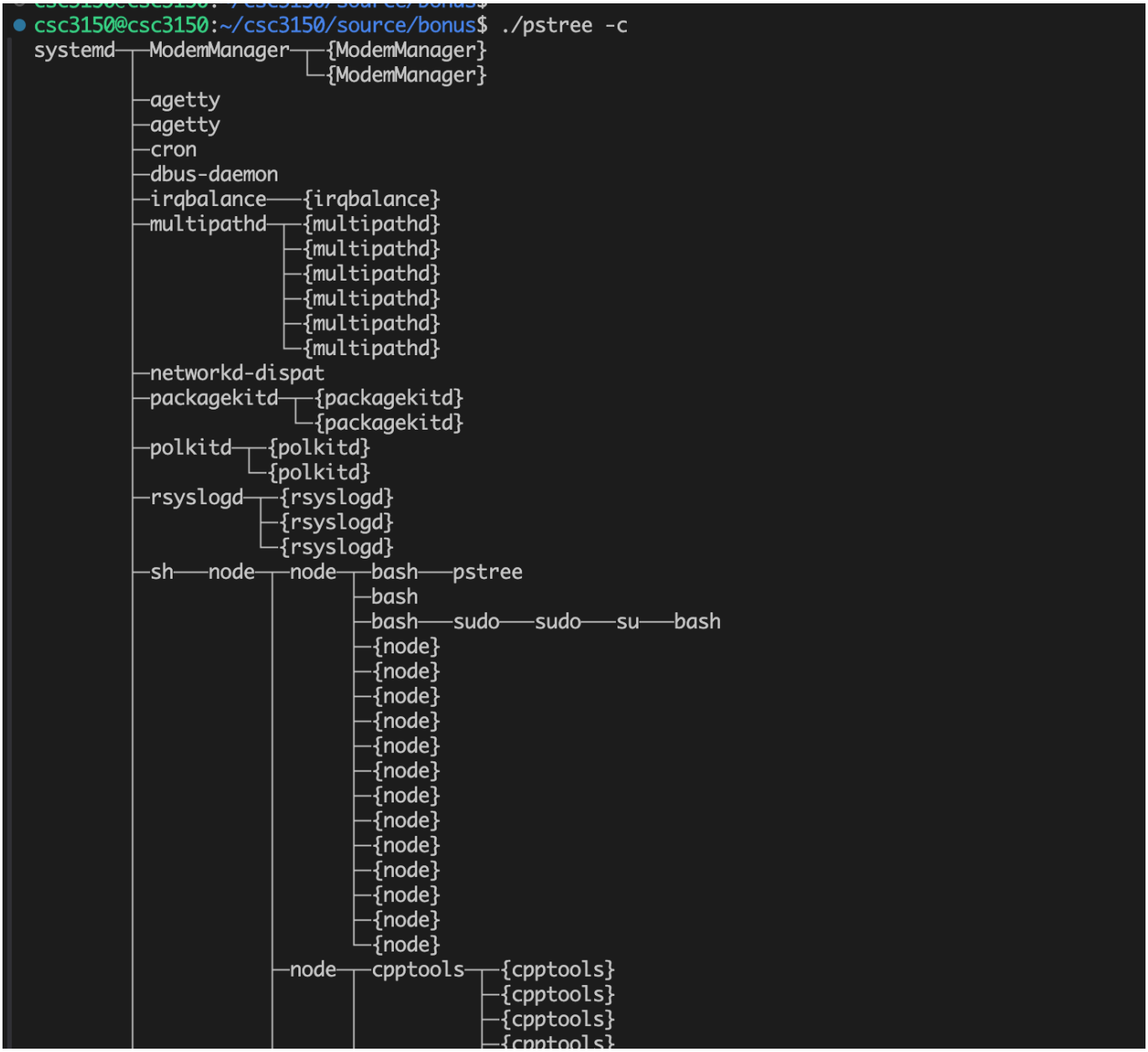
```
OPTIONS
       -a     Show command line arguments.  If the command line of a process is
swapped out, that process is shown in parentheses.  -a implicitly  disables
              compaction for processes but not threads.

       -A     Use ASCII characters to draw the tree.

       -c     Disable compaction of identical subtrees.  By default, subtrees
are compacted whenever possible.

       -n     Sort processes with the same parent by PID instead of by name.
(Numeric sort.)

       -p     Show PIDs.  PIDs are shown as decimal numbers in parentheses
after each process name.  -p implicitly disables compaction.

       -t     Show full names for threads when available.

       -T     Hide threads and only show processes.
```

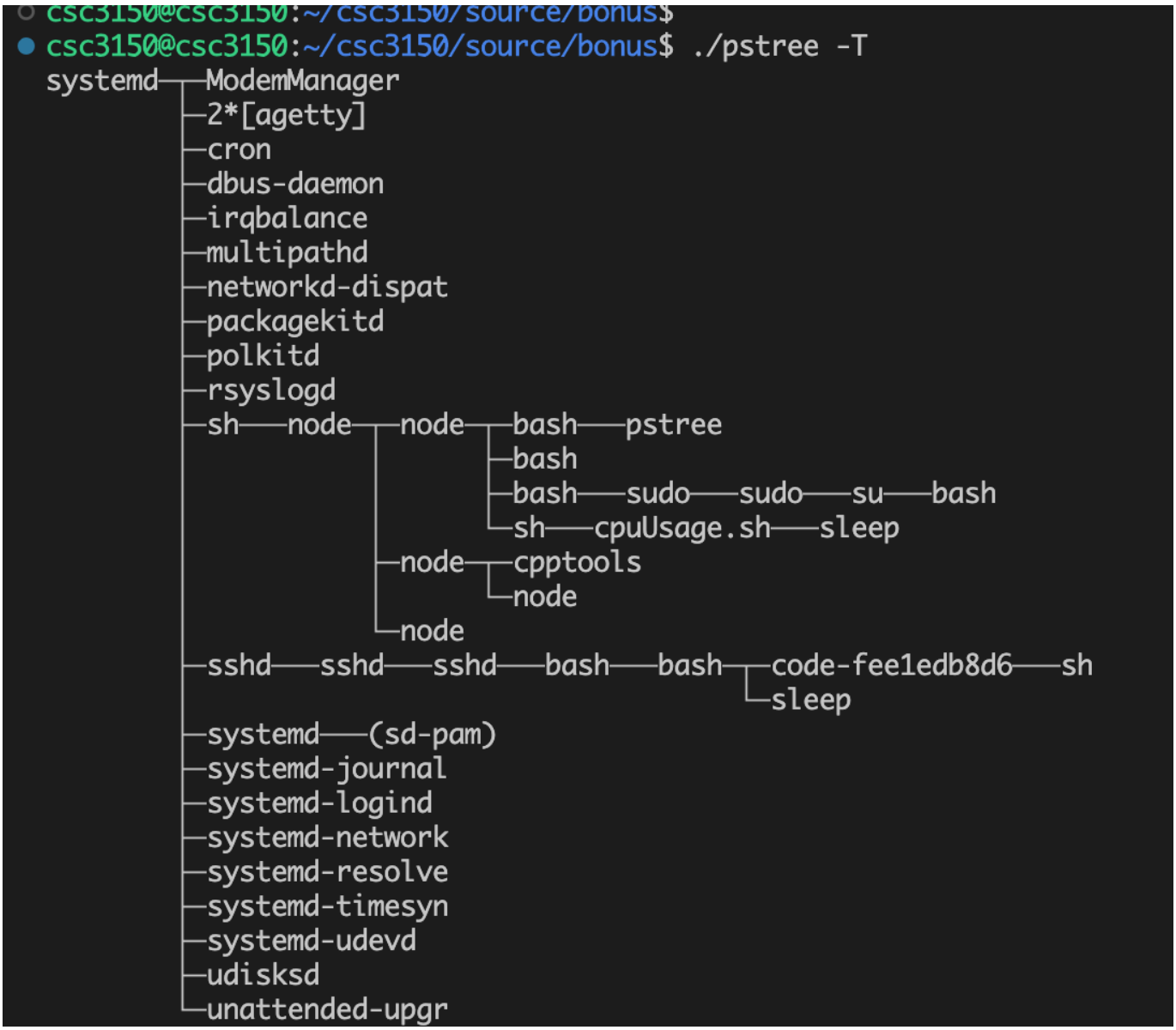
```
csc3150@csc3150:~/csc3150/source/bonus$ ./pstree
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─2*[agetty]
        ├─cron
        ├─dbus-daemon
        ├─irqbalance───{irqbalance}
        ├─multipathd───6*[{multipathd}]
        ├─networkd-dispat
        ├─packagekitd───2*[{packagekitd}]
        ├─polkitd───2*[{polkitd}]
        ├─rsyslogd───3*[{rsyslogd}]
        ├─sh───node─┬─node─┬─bash───pstree
        │           │      ├─bash
        │           │      ├─bash───sudo───sudo───su───bash
        │           │      └─13*[{node}]
        │           ├─node─┬─cpptools───8*[{cpptools}]
        │           │      ├─node───10*[{node}]
        │           │      └─13*[{node}]
        │           ├─node───12*[{node}]
        │           └─10*[{node}]
        ├─sshd───sshd───sshd───bash───bash─┬─code-fee1edb8d6─┬─sh
        │                                  │                 └─4*[{code-fee1edb8d6}]
        │                                  └─sleep
        ├─systemd───(sd-pam)
        ├─systemd-journal
        ├─systemd-logind
        ├─systemd-network
        ├─systemd-resolve
        ├─systemd-timesyn───{systemd-timesyn}
        ├─systemd-udevd
        ├─udisksd───4*[{udisksd}]
        └─unattended-upgr───{unattended-upgr}
```

```
csc3150@csc3150:~/csc3150/source/bonus$ ./pstree -a
/sbin/init
  ├─/usr/sbin/ModemManager
  │   └─2*[{ModemManager}]
  ├─agetty -o -p -- \u --noclear tty1 linux
  ├─cron -f -P
  ├─@dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
  ├─irqbalance --foreground
  │   └─{irqbalance}
  ├─multipathd -d -s
  │   └─6*[{multipathd}]
  ├─python3 /usr/bin/networkd-dispatcher --run-startup-triggers
  ├─/usr/libexec/packagekitd
  │   └─2*[{packagekitd}]
  ├─polkitd --no-debug
  │   └─2*[{polkitd}]
  ├─rsyslogd -n -iNONE
  │   └─3*[{rsyslogd}]
  ├─sh /home/csc3150/.vscode-server/cli/servers/Stable-fee1edb8d6d72a0ddff41e5f71a671c23ed924b9/server/bin/code-server --connection-token=remotessh --accept-se
rver-license-terms --start-server --enable-remote-auto-shutdown --socket-path=/tmp/code-2a56e798-8945-43
  │   └─node /home/csc3150/.vscode-server/cli/servers/Stable-fee1edb8d6d72a0ddff41e5f71a671c23ed924b9/server/out/server-main.js --connection-token=remotessh --
accept-server-license-terms --start-server --enable-remote-auto-shutdown --socket-path=/tmp/code-2a56e798-89
  │       ├─node /home/csc3150/.vscode-server/cli/servers/Stable-fee1edb8d6d72a0ddff41e5f71a671c23ed924b9/server/out/bootstrap-fork --type=ptyHost --logsPath /
home/csc3150/.vscode-server/data/logs/20241007T131312
  │       │   ├─bash --init-file /home/csc3150/.vscode-server/cli/servers/Stable-fee1edb8d6d72a0ddff41e5f71a671c23ed924b9/server/out/vs/workbench/contrib/termi
nal/browser/media/shellIntegration-bash.sh
  │       │   │   └─pstree -a
  │       │   ├─bash --init-file /home/csc3150/.vscode-server/cli/servers/Stable-fee1edb8d6d72a0ddff41e5f71a671c23ed924b9/server/out/vs/workbench/contrib/termi
nal/browser/media/shellIntegration-bash.sh
  │       │   ├─bash --init-file /home/csc3150/.vscode-server/cli/servers/Stable-fee1edb8d6d72a0ddff41e5f71a671c23ed924b9/server/out/vs/workbench/contrib/termi
nal/browser/media/shellIntegration-bash.sh
  │       │   │   └─sudo su
  │       │   │       └─sudo su
  │       │   │           └─su
  │       │   │               └─bash
  │       │   └─13*[{node}]
  │       ├─node --dns-result-order=ipv4first /home/csc3150/.vscode-server/cli/servers/Stable-fee1edb8d6d72a0ddff41e5f71a671c23ed924b9/server/out/bootstrap-for
k --type=extensionHost --transformURIs --useHostProxy=false
  │       │   ├─/home/csc3150/.vscode-server/extensions/ms-vscode.cpptools-1.21.6-linux-arm64/bin/cpptools
  │       │   │   └─8*[{cpptools}]
  │       │   ├─node /home/csc3150/.vscode-server/cli/servers/Stable-fee1edb8d6d72a0ddff41e5f71a671c23ed924b9/server/extensions/json-language-features/server/d
ist/node/jsonServerMain --node-ipc --clientProcessId=52271
  │       │   │   └─10*[{node}]
  │       │   └─13*[{node}]
  │       ├─node /home/csc3150/.vscode-server/cli/servers/Stable-fee1edb8d6d72a0ddff41e5f71a671c23ed924b9/server/out/bootstrap-fork --type=fileWatcher
  │       │   └─12*[{node}]
  │       └─10*[{node}]
  ├─sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

```
csc3150@csc3150:~/csc3150/source/bonus$ ./pstree -A
systemd-+-ModemManager---2*[{ModemManager}]
        |-2*[agetty]
        |-cron
        |-dbus-daemon
        |-irqbalance---{irqbalance}
        |-multipathd---6*[{multipathd}]
        |-networkd-dispat
        |-packagekitd---2*[{packagekitd}]
        |-polkitd---2*[{polkitd}]
        |-rsyslogd---3*[{rsyslogd}]
        |-sh---node-+-node-+-bash---pstree
        |           |      |-bash
        |           |      |-bash---sudo---sudo---su---bash
        |           |      `-13*[{node}]
        |           |-node-+-cpptools---8*[{cpptools}]
        |           |      |-node---10*[{node}]
        |           |      `-13*[{node}]
        |           |-node---12*[{node}]
        |           `-10*[{node}]
        |-sshd---sshd---sshd---bash---bash-+-code-fee1edb8d6-+-sh
        |                                  |                `-4*[{code-fee1edb8d6}]
        |                                  `-sleep
        |-systemd---(sd-pam)
        |-systemd-journal
        |-systemd-logind
        |-systemd-network
        |-systemd-resolve
        |-systemd-timesyn---{systemd-timesyn}
        |-systemd-udevd
        |-udisksd---4*[{udisksd}]
        `-unattended-upgr---{unattended-upgr}
```

```
csc3150@csc3150:~/csc3150/source/bonus$ ./pstree -p
systemd(1)─┬─ModemManager(719)───2*[{ModemManager}](749)
           ├─2*[agetty](736)
           ├─cron(675)
           ├─dbus-daemon(676)
           ├─irqbalance(682)───{irqbalance}(687)
           ├─multipathd(464)───6*[{multipathd}](472)
           ├─networkd-dispat(683)
           ├─packagekitd(17657)───2*[{packagekitd}](17659)
           ├─polkitd(684)───2*[{polkitd}](694)
           ├─rsyslogd(685)───3*[{rsyslogd}](707)
           ├─sh(1152)───node(1156)─┬─node(1262)─┬─bash(18821)───pstree(54490)
           │                       │            ├─bash(30403)
           │                       │            ├─bash(38536)───sudo(51323)───sudo(51324)───su(51325)───bash(51326)
           │                       │            └─13*[{node}](38537)
           │                       ├─node(52271)─┬─cpptools(52389)───8*[{cpptools}](52485)
           │                       │             ├─node(52422)───10*[{node}](52437)
           │                       │             └─13*[{node}](52456)
           │                       ├─node(52278)───12*[{node}](52294)
           │                       └─10*[{node}](1166)
           ├─sshd(745)───sshd(989)───sshd(1094)───bash(44877)───bash(44882)─┬─code-fee1edb8d6(44901)─┬─sh(52336)
           │                                                                │                        └─4*[{code-fee1edb8d6}](44910)
           │                                                                └─sleep(54462)
           ├─systemd(992)───(sd-pam)(993)
           ├─systemd-journal(416)
           ├─systemd-logind(696)
           ├─systemd-network(650)
           ├─systemd-resolve(652)
           ├─systemd-timesyn(615)───{systemd-timesyn}(627)
           ├─systemd-udevd(466)
           ├─udisksd(698)───4*[{udisksd}](759)
           └─unattended-upgr(746)───{unattended-upgr}(779)
```


csc3150@csc3150:~/csc3150/source/bonus$ ./pstree -c
systemd─┬─ModemManager─┬─{ModemManager}
└─{ModemManager}
├─agetty
├─agetty
├─cron
├─dbus-daemon
├─irqbalance───{irqbalance}
├─multipathd─┬─{multipathd}
├─{multipathd}
├─{multipathd}
├─{multipathd}
├─{multipathd}
└─{multipathd}
├─networkd-dispat
├─packagekitd─┬─{packagekitd}
└─{packagekitd}
├─polkitd─┬─{polkitd}
└─{polkitd}
├─rsyslogd─┬─{rsyslogd}
├─{rsyslogd}
└─{rsyslogd}
├─sh───node─┬─node─┬─bash───pstree
├─bash
├─bash───sudo───sudo───su───bash
├─{node}
├─{node}
├─{node}
├─{node}
├─{node}
├─{node}
├─{node}
├─{node}
├─{node}
├─{node}
├─{node}
└─{node}
├─node─┬─cpptools─┬─{cpptools}
├─{cpptools}
├─{cpptools}
├─{cpptools}


csc3150@csc3150:~/csc3150/source/bonus$
csc3150@csc3150:~/csc3150/source/bonus$ ./pstree -T
systemd─┬─ModemManager
├─2*[agetty]
├─cron
├─dbus-daemon
├─irqbalance
├─multipathd
├─networkd-dispat
├─packagekitd
├─polkitd
├─rsyslogd
├─sh──node─┬─node─┬─bash──pstree
│ ├─bash
│ ├─bash──sudo──sudo──su──bash
│ └─sh──cpuUsage.sh──sleep
│ ├─node─┬─cpptools
│ │ └─node
│ └─node
├─sshd──sshd──sshd──bash──bash─┬─code-fee1edb8d6──sh
│ └─sleep
├─systemd──(sd-pam)
├─systemd-journal
├─systemd-logind
├─systemd-network
├─systemd-resolve
├─systemd-timesyn
├─systemd-udevd
├─udisksd
└─unattended-upgr

What did you learn from the tasks? (2 points)

Task 1: I learn how the parent manage the child's lifecycle and the signal handling in Unix-like systems.

Task 2: I learn how to export kernel symbols and use them in a kernel module. I also learned how to create kernel threads and manage them by inserting and removing modules.

Bonus Task: I learn the structure of the `/proc` filesystem and how to read process information from it. Also explore how to build a process tree and display it in a structured way.