NYCU OS HW1 report 2021

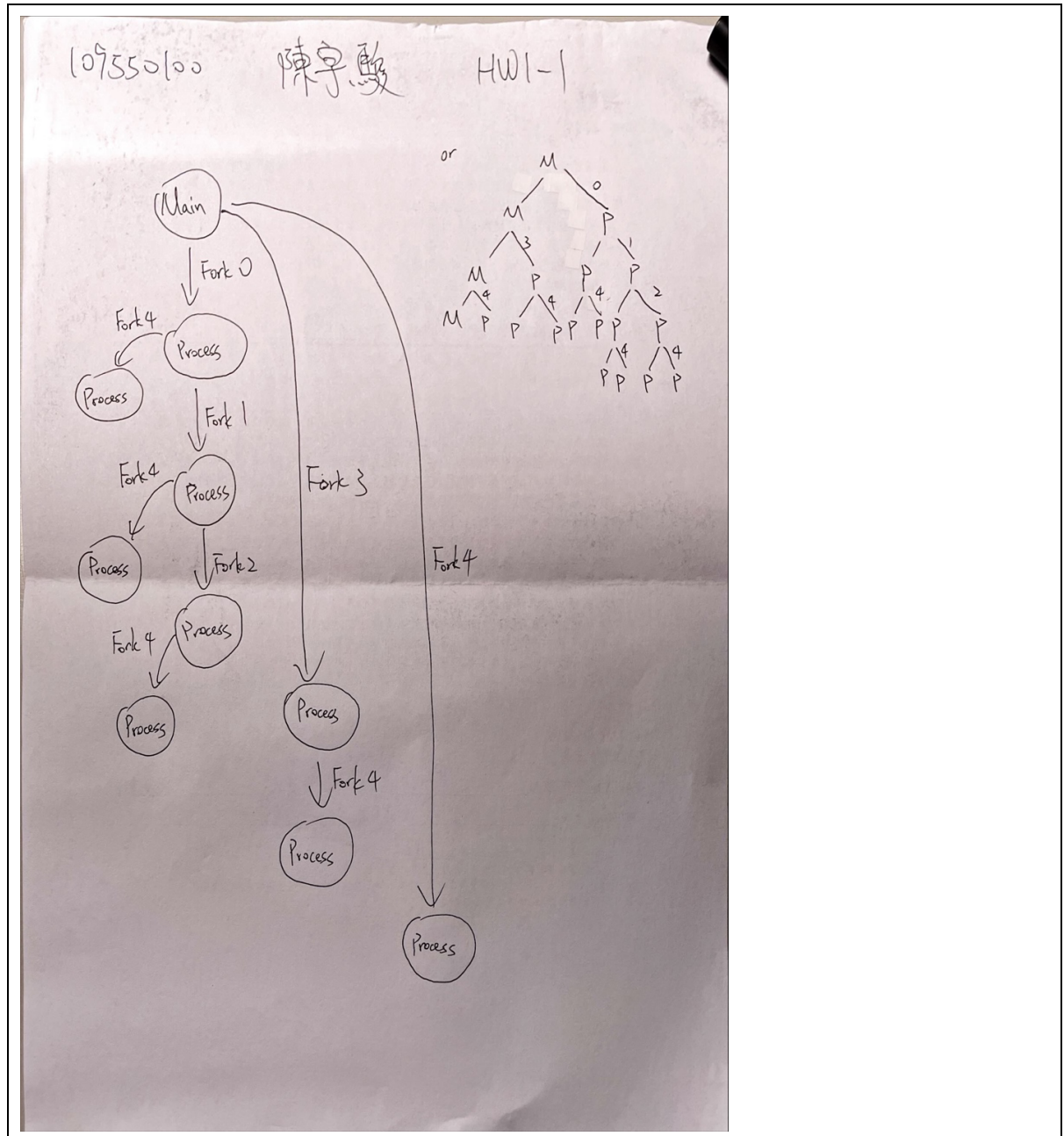Name 陳宇駿 Student Id 109550100

Q1. hw1_1

Tree format(according to OS_HW1.pdf/p.7):

Q2. hw1_2

Please put your result(screenshot).

Also, write down the problems you met and solutions.

```
Main Process ID : 27423
Fork 1. I'm the child 27434, my parent is 27423.
Fork 2. I'm the child 27435, my parent is 27434.
Fork 3. I'm the child 27436, my parent is 27435.
Fork 4. I'm the child 27437, my parent is 27423.
Fork 5. I'm the child 27438, my parent is 27437.
Fork 6. I'm the child 27439, my parent is 27423.
Fork 5. I'm the child 27440, my parent is 27439.
Fork 5. I'm the child 27441, my parent is 27423.
```

Problems & solutions:

I spent a lot of time figuring out if there is no "if (pid == 0)

{ /*do something*/ }" block, child process will still be

spawned or not.   Though this problem sounds silly, solving

such confusion makes me feel more steady while counting

how many processes have been forked.

Q3. hw1-3

Please put your result(screenshot).

```
> ./a.out
osh>cat test.txt
test content!!
osh>ls
Homework 1.pdf  OS_report.docx  a.out          hw1_1.c       hw1_2.cpp     hw1_3.cpp     test.txt      ~$_report.docx
osh>date
2022年10月 3日 週一 00時14分02秒 CST
osh>ps
  PID TTY          TIME CMD
27740 ttys000    0:00.03 ./a.out
73820 ttys000    0:04.93 /bin/zsh -il
73868 ttys000    0:00.01 /bin/zsh -il
73882 ttys000    0:00.71 /Users/shflte/.cache/gitstatus/gitstatusd-darwin-x86_64 -G v1.3.1 -s 1 -u 1 -d 1 -c 1 -m -1 -v FATAL -t 16
osh>ps -f
  UID   PID  PPID   C STIME   TTY          TIME CMD
  501 27740 73820   0 12:13上午 ttys000    0:00.03 ./a.out
  501 73820 20106   0 7:07下午 ttys000    0:04.93 /bin/zsh -il
  501 73868     1   0 7:07下午 ttys000    0:00.01 /bin/zsh -il
  501 73882 73868   0 7:07下午 ttys000    0:00.71 /Users/shflte/.cache/gitstatus/gitstatusd-darwin-x86_64 -G v1.3.1 -s 1 -u 1 -d 1 -c 1 -m -1 -v FATAL -t 16
osh>pwd
/Users/shflte/Desktop/OS/Homework 1
osh>exit
~/Desktop/OS/Homework 1 28s >
```

Problems & solutions:

This hw is relatively easier than hw1_2, since this hw does not involve a lot of complicated forking structure.   As long as I figured out how to pass the exit status from child process to parent process, the hw was done right away.