**CS149**

**HW2**

**Jooyul Yoon**

**014597154**

Q1.

Chart

Description automatically generated**Total number of unique process:**

p1 – the parent process

p2 – p1 created child process by fork() in line 2



p3 – p2 created child process by fork() in line 4



p4 – p2 created child process by fork() in line 6

p5 – p3 created child process by fork() in line 6



p6 – p1 created child process by fork() in line 9

p7 – p2 created child process by fork() in line 9



p8 – p3 created child process by fork() in line 9

p9 – p4 created child process by fork() in line 9

p10 – p5 created child process by fork() in line 9



Thus, there are 10 unique processes and 7 threads.

**Total number of threads:**

Since I got 7 pthread\_create() calls, there are 7 unique threads.

Text

Description automatically generatedQ2. a

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screen, screenshot

Description automatically generated

Text

Description automatically generated

Q2. b - “29 rows with 5 threads”

29 / 5 = 5 … 4, where 29 is total number of rows, 5 is number thread, the other 5 is divisor, and 4 is remainder. I need to make first 4 threads with 6 rows, and the last thread with 5 rows. In another words, each thread has at least 5 rows(divisor), but the first 4(remainder) threads have one more row.

In this case, the five threads compute row 0 - 5, 6 – 11, 12 – 17, 18 – 23, 24 – 28. It was important to know the first row of each thread. Therefore, I initialized it zero and updated the value in the end of each thread for its next thread.

Pseudo code:

*Int first\_row = 0*

*for each thread*

*{*

*If remainder > 0*

*{*

*# of rows = divisor + 1*

*Decrement remainder by 1*

*}*

*Else*

*# of rows = divisor*

*First\_row += number of rows of current thread*

*}*