

CS149
HW2
Jooyul Yoon
014597154

Q1.

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

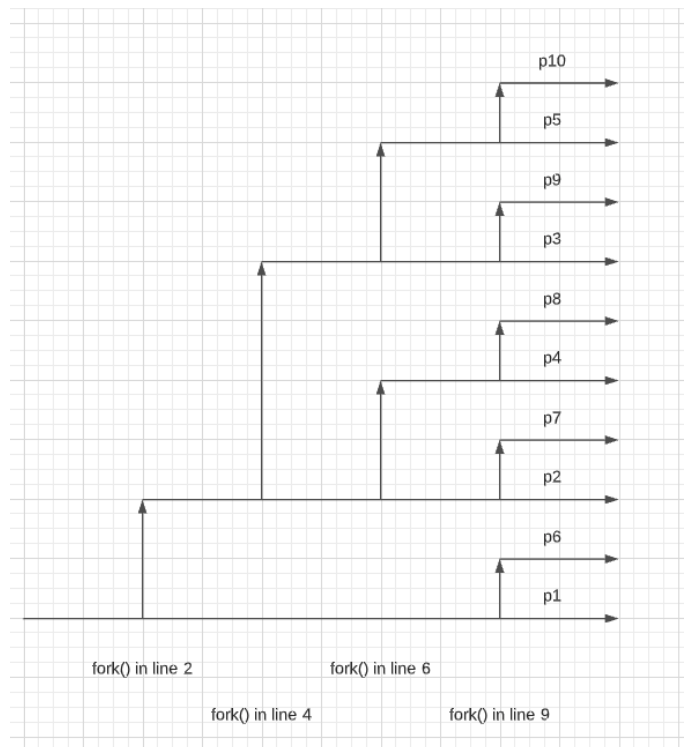
p7 – p2 created child process by fork() in line 9

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 6 unique processes in total.



Total number of threads:

Since I got 7 pthread_create() calls, there are 7 unique threads.

Q2. a

```
~/Documents/SJSU/CS149/hw2 [ ] ./matrix_multiply 0
CS149 Spring 2021 parallel matrix multiplication from Jooyul Yoon

thread[0]: row= 0, col= 0
thread[1]: row= 0, col= 1
thread[2]: row= 0, col= 2
thread[3]: row= 1, col= 0
thread[4]: row= 1, col= 1
thread[5]: row= 1, col= 2
thread[6]: row= 2, col= 0
thread[7]: row= 2, col= 1
thread[8]: row= 2, col= 2
thread[9]: row= 3, col= 0
thread[10]: row= 3, col= 1
thread[11]: row= 3, col= 2
thread[12]: row= 4, col= 0
thread[13]: row= 4, col= 1
thread[14]: row= 4, col= 2

--- Resultant Matrix ---

73      50      93
211     152     231
349     254     369
487     356     507
625     458     645
```

```
~/Documents/SJSU/CS149/hw2 [ ] ./matrix_multiply 1
CS149 Spring 2021 parallel matrix multiplication from Jooyul Yoon

thread[0] - first row= 0, number of rows = 5

--- Resultant Matrix ---

73      50      93
211     152     231
349     254     369
487     356     507
625     458     645
```

```
~/Documents/SJSU/CS149/hw2 [ ] ./matrix_multiply 2
CS149 Spring 2021 parallel matrix multiplication from Jooyul Yoon

thread[0] - first row= 0, number of rows = 3
thread[1] - first row= 3, number of rows = 2

--- Resultant Matrix ---

73      50      93
211     152     231
349     254     369
487     356     507
625     458     645
```

```
~/Documents/SJSU/CS149/hw2 [ ] ./matrix_multiply 4
CS149 Spring 2021 parallel matrix multiplication from Jooyul Yoon

thread[0] - first row= 0, number of rows = 2
thread[1] - first row= 2, number of rows = 1
thread[2] - first row= 3, number of rows = 1
thread[3] - first row= 4, number of rows = 1

--- Resultant Matrix ---

73      50      93
211     152     231
349     254     369
487     356     507
625     458     645
```

Q2. b - "29 rows with 5 threads"

$29 / 5 = 5 \dots 4$, 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

}