CS 481

PROGRAMMING ASSIGNMENT 2

BAKER, ALEX

Problem 1

Part A

```
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY       TIME     CMD
0 S  4424 27826 27825  0  80   0 -  5065 wait     pts/2    00:00:00 bash
0 S  4424 28840 27826  0  80   0 -  2859 hrtime pts/2    00:00:00 tail
0 S  4424 28965 27826  0  80   0 -  2859 hrtime pts/2    00:00:00 tail
0 S  4424 29011 27826  0  80   0 -  2859 hrtime pts/2    00:00:00 tail
0 R  4424 29027 27826  0  80   0 -  3554 -          pts/2    00:00:00 ps
```

1. 27826, bash, sleeping
2. 28840, tail, sleeping
3. 28965, tail, sleeping
4. 29011, tail, sleeping
5. 29027, ps, running

Part B

```
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0     1     0  0  80   0 -  9197 -      ?        00:00:05 init
1 S     0     2     0  0  80   0 -     0 -      ?        00:00:00 kthreadd
1 S     0     3     2  0  80   0 -     0 -      ?        00:00:00 ksoftirqd/0
1 S     0     5     2  0  60 -20 -     0 -      ?        00:00:00 kworker/0:0H
1 S     0     7     2  0  80   0 -     0 -      ?        00:01:00 rcu_sched
1 S     0     8     2  0  80   0 -     0 -      ?        00:01:51 rcuos/0
1 S     0     9     2  0  80   0 -     0 -      ?        00:01:56 rcuos/1
```

1. 1, init, sleeping
2. 2, kthread, sleeping
3. 7, rcu_sched, sleeping

Part c

Trace: Bash

1. 1: init
2. 1734: sshd
3. 27770: sshd: alexebaker [priv]
4. 27825: sshd: alexebaker@pts/
5. 27826: bash

Depth: 5

Problem 2

Output

```
from C1: own PID=87757, parent's PID=87756
from C2: own PID=87758, parent's PID=87756
from P0: own PID=87756, PID of C1=87757, PID of C2=87758, total elapsed time in
milliseconds=0.3180
```

Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>
#include <sys/types.h>


int fib(int x);


int main(int argc, char *argv[])
{
   pid_t pid, ppid;
   int numChildren = 2;
   int cpids[numChildren];
   int i = 0;
   clock_t start, end;
   double time;

   start = clock();
   ppid = getpid();
   for(i = 0; i < numChildren; i++)
   {
      pid = fork();
      if (pid < 0)
      {
         fprintf(stderr, "Fork Failed");
         exit(1);
      }
      else if (pid == 0)
      {
         printf("from C%d: own PID=%d, parent's PID=%d\n", i+1, getpid(), ppid);
```

```
            fib(20);
            exit(0);
        }
        else
        {
            cpids[i] = pid;
            wait(NULL);
        }
    }
    end = clock();

    printf("from P0: own PID=%d", ppid);
    for (i = 0; i < numChildren; i++)
    {
        printf(", PID of C%d=%d", i+1, cpids[i]);
    }
    time = ((double)(end - start) / CLOCKS_PER_SEC) * 1000;
    printf(", total elapsed time in milliseconds=%.4f\n", time);
    return 0;
}


int fib(int x)
{
    int i = 0;
    int rint = rand() % 30;
    double dummy;

    for (i = 0; i < rint*100; i++)
    {
        dummy = (2.345 * i * 8.765) / 1.234;
    }

    if (x == 0)
    {
        return 0;
    }
    else if (x == 1)
    {
        return 1;
    }
    else
    {
        return fib(x-1) + fib(x-2);
    }
}
```

Problem 3

Output

```
from C1: own PID=4523, parent's PID=4522
Tue Sep 20 19:24:38 MDT 2016
from C2: own PID=4524, parent's PID=4522
aebaker  console  Sep 12 17:03
aebaker  ttys000  Sep 13 13:37
aebaker  ttys001  Sep 19 13:16
aebaker  ttys003  Sep 19 13:22
aebaker  ttys005  Sep 20 12:29
from P0: own PID=4522, PID of C1=4523, PID of C2=4524, total elapsed time in
milliseconds=0.3980
```

Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>
#include <sys/types.h>


int fib(int x);


int main(int argc, char *argv[])
{
    pid_t pid, ppid;
    int numChildren = 2;
    int cpids[numChildren];
    int i = 0;
    clock_t start, end;
    double time;

    start = clock();
    ppid = getpid();
    for(i = 0; i < numChildren; i++)
    {
        pid = fork();
        if (pid < 0)
        {
            fprintf(stderr, "Fork Failed");
            exit(EXIT_FAILURE);
        }
        else if (pid == 0)
        {
            printf("from C%d: own PID=%d, parent's PID=%d\n", i+1, getpid(), ppid);
            fib(20);

            if (i == 0)
            {
                execl("/bin/date", "date", NULL);
            }
            else if (i == 1)
            {
                execl("/usr/bin/who", "who", NULL);
            }
            exit(EXIT_SUCCESS);
        }
        else
        {
            cpids[i] = pid;
            wait(NULL);
        }
    }
    end = clock();

    printf("from P0: own PID=%d", ppid);
    for (i = 0; i < numChildren; i++)
    {
        printf(", PID of C%d=%d", i+1, cpids[i]);
    }
    time = ((double)(end - start) / CLOCKS_PER_SEC) * 1000;
    printf(", total elapsed time in milliseconds=%.4f\n", time);
    return 0;
}


int fib(int x)
{
    int i = 0;
    int rint = rand() % 30;
    double dummy;

    for (i = 0; i < rint*100; i++)
    {
        dummy = (2.345 * i * 8.765) / 1.234;
    }

    if (x == 0)
    {
        return 0;
    }
    else if (x == 1)
    {
        return 1;
    }
    else
    {
        return fib(x-1) + fib(x-2);
    }
}
```