OS Project 2566

There are 3 assignments here. A1 is about Docker and UNIX/Linux commands, while A2 and A3 are about programming fork() system call.

Assignment 3: Docker

ให้เขียน dockerfile เพื่อสร้าง image ชื่อภาษาอังกฤษของท่านเอง เพื่อทำงานต่อไปนี้

1. ใช้ Ubuntu รุ่นล่าสุด แล้วใช้งาน bash เมื่อเริ่มต้น โดยก่อนเริ่มใช้ bash ให้ทำดังต่อไปนี้
2. **ส่ง**โค้ดภาษา C ชื่อ **`csine.c`** เพื่อทดสอบโปรแกรมง่ายๆ จากเครื่อง นศ. เข้าไปใน home directory ด้วย โดยโปรแกรมนี้ แสดงค่า **`sin(x)`** โดย x ไล่จาก -2π ถึง 2π โดยเพิ่มทีละ π/3 และแสดงเป็น 2 คอลัมน์ ใช้จุดทศนิยม 3 ตำแหน่งทั้ง 2 คอลัมน์ ผลลัพธ์จะแสดงดังต่อไปนี้:

```
-6.283  0.000
-5.236  0.866
-4.189  0.866
-3.142  0.000
-2.094  -0.866
-1.047  -0.866
-0.000  -0.000
1.047   0.866
2.094   0.866
3.142   0.000
4.189   -0.866
5.236   -0.866
6.283   -0.000
```

3. **ส่ง**โค้ดภาษา Python ชื่อ **`np2pd.py`** ดังต่อไปนี้ เข้าไปเพื่อทดสอบ

```
# https://www.geeksforgeeks.org/create-a-dataframe-from-a-numpy-array-and-specify-the-index-column-and-column-headers/
# importiong the modules
import pandas as pd
import numpy as np

# creating the Numpy array
array = np.array([[1, 1, 1], [2, 4, 8], [3, 9, 27], [4, 16, 64],
                  [5, 25, 125], [6, 36, 216], [7, 49, 343]])

# creating a list of index names and a list of column names
index_values = ['first', 'second', 'third', 'fifth', 'sixth', 'seventh']
column_values = ['number', 'squares', 'cubes']

# creating and then displaying the dataframe
df = pd.DataFrame(data = array,
                  index = index_values,
                  columns = column_values)
print(df)
```

4. หากใช้ Ubuntu ปกติใน Docker พบว่า จะยังไม่มีโปรแกรมจำเป็นหลายตัวให้ เช่น Python, Editor, C Compiler, sudo ดังนั้น ให้สร้าง dockerfile เพื่อให้จัดการติดตั้งโปรแกรมเหล่านี้ให้โดยอัตโนมัติ

เมื่อเข้าไปใน bash shell เริ่มต้นแล้ว ให้ทำดังต่อไปนี้

5. คอมไพล์ **`csine.c`** ข้างต้น ให้ได้โปรแกรมชื่อ hello (ไม่ใช่ **`a.out`**) แล้วสั่งให้ทำงานเพื่อทดสอบ โดยการเรียก **`csine`** ตรงๆ

6. รันทดสอบ **`np2pd.py`** ที่นำเข้าไปข้างต้นให้สำเร็จ จะได้ผลลัพธ์ดังต่อไปนี้

```
         number  squares  cubes
first         1        1      1
second        2        4      8
third         3        9     27
fourth        4       16     64
fifth         5       25    125
sixth         6       36    216
seventh       7       49    343
```

7. ดาวน์โหลดไฟล์ https://files.grouplens.org/datasets/movielens/ml-latest-small.zip เข้ามา

8. Uncompress ไฟล์ดังกล่าว ออกมาจะได้เป็น 1 subdirectory ข้างในนั้นจะพบไฟล์ ratings.csv ซึ่งมีรูปแบบดังนี้

   `userId,movieId,rating,timestamp`

   เช่น

```
1,1256,5.0,964981442
1,1258,3.0,964983414
1,1265,4.0,964983599
1,1270,5.0,964983705
608,6888,1.5,1117675045
608,7004,0.5,1117506252
610,2332,1.5,1493849039
```

9. ให้เขียน shell commands/script เพื่อรายงานจำนวน ratings ที่ให้คะแนน 5.0, 4.0, 1.0, 0.5 ตามลำดับ

10. ให้เขียน shell commands/script เพื่อคัดกรอง ratings ที่คะแนน 0.5 และให้เรียงตาม movieID (จากน้อยไปมาก) แล้วบันทึกลงไฟล์ จากนั้น ให้แสดงส่วนต้นของไฟล์นั้นมา 15 บรรทัด และส่วนท้ายมา 15 บรรทัด ผลลัพธ์ควรจะได้ดังต่อไปนี้

```
76,1,0.5,1439165548
298,2,0.5,1450452897
308,3,0.5,1421374465
490,5,0.5,1324370305
517,10,0.5,1487957717
112,17,0.5,1513989967
517,17,0.5,1487953834
3,31,0.5,1306463578
112,36,0.5,1513989966
112,39,0.5,1513989927
608,44,0.5,1117504562
104,47,0.5,1053336550
426,47,0.5,1451081886
608,48,0.5,1117161754
76,48,0.5,1439168949
```

```
111,165645,0.5,1516156050
567,166461,0.5,1525288081
111,166534,0.5,1516141946
550,167296,0.5,1488728333
153,172547,0.5,1525553047
153,173145,0.5,1525553026
21,173307,0.5,1500701120
153,175303,0.5,1525553022
184,175475,0.5,1537109570
50,175485,0.5,1514240073
153,179819,0.5,1525553024
380,179819,0.5,1536872721
567,184253,0.5,1525289944
153,184471,0.5,1525553051
184,184641,0.5,1537094808
```

Assignment 2: Fork

ให้ปรับโค้ดในเอกสารการสอนต่อไปนี้เพื่อให้ทำงานได้จริงบน Ubuntu Docker ที่สร้างขึ้น แต่ปรับให้ child process ไปทำงานคำสั่ง ps แทน และปรับการแสดงผลให้เป็นไปตามภาพด้านล่าง

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int pid;
    /* create another process */
    pid = fork();
    if (pid < 0) { /* error occurred */
           fprintf(stderr, "Fork Failed");
           exit(-1);
    }
    else if (pid == 0) { /* child process */
           execlp("./child","child",NULL);
    }
    else { /* parent process waits for */
           /* the child to complete */
           wait(NULL);
           printf("Child completed");
           exit(0);
    }
}
```

```
root@chukiat:~# afork
Parent: PID 6230 and PPID 6092, and waiting my child process to terminate.
I'm the child with PID 6231 and PPID 6230, about to call ps using execlp
after 3 sec.
  PID TTY          TIME CMD
 6092 pts/3    00:00:00 bash
 6230 pts/3    00:00:00 afork
 6231 pts/3    00:00:00 ps
Parent: I see my child completed.
root@chukiat:~#
```

ในรูปนี้ โพรเซสแม่ที่รันโปรแกรม มี Process ID = 6230 และ เห็น Process ID ของ parent มัน (ซึ่งก็คือ bash) เท่ากับ 6092. ส่วนโพรเซสลูกที่ถูกสร้างขึ้นก็แสดง Process ID ของตัวมันเองและแม่ของมันด้วยเช่นกัน โพรเซสลูกจะนอนหลับ 3 วินาทีก่อนจะสิ้นสุดการทำงาน และในระหว่างนั้นโพรเซสแม่จะรอจนกว่าลูกจะจบการทำงานจึงแสดงข้อความออกมาว่า “Parent: I see my child completed.”

Assignment 3: Multi-Fork

ให้ปรับโค้ดในข้อที่แล้วให้โพรเซสแม่สร้างโพรเซสลูกจำนวนหลายตัว (MAX) อย่างต่อเนื่องห่างกันครั้งละ 1 วินาที จากนั้นรอดูโพรเซสลูกสิ้นสุดการทำงานและนับด้วย ส่วนโพรเซสลูกแต่ละตัวให้แสดงเลขที่ Process ID ของตัวเองและของ its parent ออกมาดังรูปต่อไปนี้ จากนั้นจะรอเป็นเวลา PROCTIME = 30 วินาที ก่อนจะสิ้นสุดการทำงาน. ทั้งนี้...

1. ให้กำหนดพารามิเตอร์ MAX = 20 และ PROCTIME = 30 ไว้ต้นโปรแกรมเพื่อให้ปรับได้สะดวก
2. เปิดอีก terminal หนึ่งเพื่อสังเกตจำนวนโพรเซสลูกว่าเพิ่มหรือลดอย่างไร ให้รายงาน

เมื่อรับโปรแกรมแล้วจะได้ผลทำนองต่อไปนี้...

```
Child( 0) PID 6260       PPID 6259, about to terminate in 30 sec.
Child( 1) PID 6261       PPID 6259, about to terminate in 30 sec.
Child( 2) PID 6262       PPID 6259, about to terminate in 30 sec.
Child( 3) PID 6263       PPID 6259, about to terminate in 30 sec.
Child( 4) PID 6264       PPID 6259, about to terminate in 30 sec.
Child( 5) PID 6266       PPID 6259, about to terminate in 30 sec.
Child( 6) PID 6267       PPID 6259, about to terminate in 30 sec.
Child( 7) PID 6268       PPID 6259, about to terminate in 30 sec.
Child( 8) PID 6269       PPID 6259, about to terminate in 30 sec.
Child( 9) PID 6270       PPID 6259, about to terminate in 30 sec.
Child(10) PID 6271       PPID 6259, about to terminate in 30 sec.
Child(11) PID 6272       PPID 6259, about to terminate in 30 sec.
Child(12) PID 6273       PPID 6259, about to terminate in 30 sec.
Child(13) PID 6274       PPID 6259, about to terminate in 30 sec.
Child(14) PID 6275       PPID 6259, about to terminate in 30 sec.
Child(15) PID 6276       PPID 6259, about to terminate in 30 sec.
Child(16) PID 6277       PPID 6259, about to terminate in 30 sec.
Child(17) PID 6278       PPID 6259, about to terminate in 30 sec.
Child(18) PID 6279       PPID 6259, about to terminate in 30 sec.
Child(19) PID 6280       PPID 6259, about to terminate in 30 sec.
Parent: I see my child #0 completed.
Parent: I see my child #1 completed.
Parent: I see my child #2 completed.
Parent: I see my child #3 completed.
Parent: I see my child #4 completed.
Parent: I see my child #5 completed.
Parent: I see my child #6 completed.
Parent: I see my child #7 completed.
Parent: I see my child #8 completed.
Parent: I see my child #9 completed.
Parent: I see my child #10 completed.
Parent: I see my child #11 completed.
Parent: I see my child #12 completed.
Parent: I see my child #13 completed.
Parent: I see my child #14 completed.
Parent: I see my child #15 completed.
Parent: I see my child #16 completed.
Parent: I see my child #17 completed.
Parent: I see my child #18 completed.
Parent: I see my child #19 completed.
```

```
top - 03:57:53 up  2:20,  0 users,  load average: 0.00, 0.00, 0.00
Tasks:  21 total,   1 running,  20 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.1 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0
st
MiB Mem :   7817.3 total,   4293.4 free,   1072.1 used,   2451.8 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   6470.8 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root      20   0    4620   3872   3288 S   0.0   0.0   0:00.07 bash
   39 root      20   0    4620   3912   3320 S   0.0   0.0   0:00.12 bash
 5941 root      20   0    4620   3904   3312 S   0.0   0.0   0:00.09 bash
 6085 root      20   0    4916   4224   2644 S   0.0   0.1   0:00.01 nano
 6092 root      20   0    4620   3900   3308 S   0.0   0.0   0:00.06 bash
 6259 root      20   0    2636    980    888 S   0.0   0.0   0:00.00 forks
 6260 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6261 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6262 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6263 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6264 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6265 root      20   0    7308   3400   2832 R   0.0   0.0   0:00.00 top
 6266 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6267 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6268 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6269 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6270 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
 6271 root      20   0    2768     92      0 S   0.0   0.0   0:00.00 forks
```

Code Hint:

```python
import math

x = -2 * math.pi

while(x <= 2 * math.pi):
   print(round(x, 3), '\t', roundmath.sin(x, 3))
   x += math.pi / 3
```

```c
#include <stdio.h>
#include <math.h>

#define PI 3.14159265358979323846

int main() {
    double x;
    printf("x\tsin(x)\n");

    for (x = -2.0 * PI; x <= 2.0 * PI; x += PI/3.0)
        printf("%.3lf\t%.3lf\n", x, sin(x));
    return 0;
}
```

```
# Use the official Ubuntu as the base image
FROM ubuntu:latest

COPY *.c    /root
COPY *.py   /root
COPY *.html /root

# Update the package lists, install essential packages, and clean up
RUN apt update \
    && apt install -y sudo gcc python3 wget nano p7zip p7zip-full zip unzip  \
    && rm -rf /var/lib/apt/lists/*

# Add the current directory (.) to the PATH environment variable
ENV PATH="${PATH}:."

# Set the working directory
WORKDIR /root

# Start a Bash shell when the container is run
CMD ["/bin/bash"]
```