

# MIDTERM EXAM

- 다음 문제들에 대해 소스코드파일과 실행결과캡처파일을 zip 으로 압축하여 첨부파일로 제출. 파일명: 학번\_이름.zip
  - 제출기한 17:00 을 넘긴 경우 모두 0 점 처리하므로 제출기한을 엄수할 것.
1. HW1 에서 작성한 파이썬 스크립트를 새로운 유저를 생성하여 새로운 유저의 이름으로 실행하고자 한다. 새로운 유저를 생성하는 과정을 셸 스크립트로 작성하여라. 셸 스크립트 파일, 파이썬 스크립트 파일, 결과 실행화면을 제출하여라. (25 점)

참고 1: 셸스크립트는 아래를 참고하여 작성하여라.

```
#!/bin/bash

# 유저생성 (ID: iot 본인학번. ex) iot20171234)

# 생성유저에게 권한부여

# pythonTEST.py 파일을 새로운 유저의 홈디렉터리로 복사

# 새로운 유저로 변경
```

참고 2: 실행화면

## -셸스크립트를 통한 유저 생성 및 변경 화면

```
pi@raspberrypi:~$ sh iot_TEST
Adding user 'iot20180240' ...
Adding new group 'iot20180240' (1001) ...
Adding new user 'iot20180240' (1001) with group 'iot20180240' ...
Creating home directory '/home/iot20180240' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for iot20180240
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
Password:
iot20180240@raspberrypi:/home/pi $
```

## -파이썬스크립트 실행 화면

```
iot20180240@raspberrypi:/home/pi $ cd
iot20180240@raspberrypi:~$ python3 pythonTEST.py
Starting the GPIO TEST Python Script!
There is an incorrect number of arguments
  usage is: pythonTEST.py command
  where command is one of setup, on, off, status, or close
iot20180240@raspberrypi:~$ python3 pythonTEST.py setup
Starting the GPIO TEST Python Script!
Setting up the TEST GPIO
End of Python script
iot20180240@raspberrypi:~$ python3 pythonTEST.py on
Starting the GPIO TEST Python Script!
Turning the TEST on
End of Python script
iot20180240@raspberrypi:~$ python3 pythonTEST.py status
Starting the GPIO TEST Python Script!
Getting the TEST state value
1
End of Python script
iot20180240@raspberrypi:~$ python3 pythonTEST.py off
Starting the GPIO TEST Python Script!
Turning the TEST off
End of Python script
iot20180240@raspberrypi:~$ python3 pythonTEST.py status
Starting the GPIO TEST Python Script!
Getting the TEST state value
0
End of Python script
iot20180240@raspberrypi:~$ python3 pythonTEST.py close
Starting the GPIO TEST Python Script!
Closing down the TEST GPIO
End of Python script
iot20180240@raspberrypi:~$
```

## 2. 라즈베리파이에서 다음을 수행하여라.

```
pi@raspberrypi:~ $ cd
pi@raspberrypi:~ $ mkdir Test
pi@raspberrypi:~ $ cd Test
pi@raspberrypi:~ $ touch file{1..100}
```

Test 디렉터리에 있는 file1 부터 file100 까지 100 개 파일의 이름을 001, 002, ... , 100 으로 바꾸는 셸 스크립트를 작성하여라. 셸 스크립트 파일, 결과 실행화면을 제출하여라. 만약 파일이 1000 개가 있으면 0001, 0002, ... , 1000 으로 이름이 바뀌도록 일반화된 스크립트를 작성해야 한다. (25 점)

참고: 실행화면

```
pi@raspberrypi:~/Test $ ls
file1  file2  file30  file41  file52  file63  file74  file85  file96
file10  file20  file31  file42  file53  file64  file75  file86  file97
file100  file21  file32  file43  file54  file65  file76  file87  file98
file11  file22  file33  file44  file55  file66  file77  file88  file99
file12  file23  file34  file45  file56  file67  file78  file89  rename.sh
file13  file24  file35  file46  file57  file68  file79  file9
file14  file25  file36  file47  file58  file69  file8  file90
file15  file26  file37  file48  file59  file7  file80  file91
file16  file27  file38  file49  file6  file70  file81  file92
file17  file28  file39  file5  file60  file71  file82  file93
file18  file29  file4  file50  file61  file72  file83  file94
file19  file3  file40  file51  file62  file73  file84  file95
pi@raspberrypi:~/Test $ ./rename.sh
pi@raspberrypi:~/Test $ ls
001 008 015 022 029 036 043 050 057 064 071 078 085 092 099
002 009 016 023 030 037 044 051 058 065 072 079 086 093 100
003 010 017 024 031 038 045 052 059 066 073 080 087 094 rename.sh
004 011 018 025 032 039 046 053 060 067 074 081 088 095
005 012 019 026 033 040 047 054 061 068 075 082 089 096
006 013 020 027 034 041 048 055 062 069 076 083 090 097
007 014 021 028 035 042 049 056 063 070 077 084 091 098
pi@raspberrypi:~/Test $
```

3. 아래 실행화면을 참고하여 함수포인터배열을 활용하여 계산기 프로그램을 구현하여라.  
소스코드 파일, 결과 실행화면을 제출하여라. (25 점)

참고: 실행화면

```
pi@raspberrypi:~ $ ./calc
Enter your choice: 1 for sum, 2 for subtraction, 3 for mult, 4 for division, and 0 for exit: 1
Enter the two numbers: 1.2 3.4
4.600000
Enter your choice: 1 for sum, 2 for subtraction, 3 for mult, 4 for division, and 0 for exit: 2
Enter the two numbers: 4.5 6.7
-2.200000
Enter your choice: 1 for sum, 2 for subtraction, 3 for mult, 4 for division, and 0 for exit: 3
Enter the two numbers: 8.9 0.1
0.890000
Enter your choice: 1 for sum, 2 for subtraction, 3 for mult, 4 for division, and 0 for exit: 4
Enter the two numbers: 2.3 4.5
0.511111
Enter your choice: 1 for sum, 2 for subtraction, 3 for mult, 4 for division, and 0 for exit: 0
pi@raspberrypi:~ $
```

4. 다음은 결과를 얻기까지 긴 시간이 걸리는 프로그램의 소스코드이다. 2 개 이상의 스레드를 사용하도록 소스코드를 수정하여 이 프로그램의 실행시간을 단축시켜라. 소스코드 파일, 결과 실행화면을 제출하여라. (25 점)

참고 1: test\_thread\_single.c

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

int main (int argc, char **argv)
{
    long long a = 0;

    for (int i=1; i<=999999990; i++)
        a =a +i;

    printf("Value: %lld\n", a);
}
```

참고 2: 실행화면

```
pi@raspberrypi:~ $ time ./test_thread_single
Value: 499999990500000045
real    0m28.418s
user    0m28.382s
sys     0m0.010s
pi@raspberrypi:~ $ gcc test_thread.c -o test_thread -pthread
pi@raspberrypi:~ $ time ./test_thread
Value: 499999990500000045
real    0m12.461s
user    0m36.999s
sys     0m0.010s
pi@raspberrypi:~ $
```