

実験番号 4

実験題目 UNIX 入門Ⅲ

実験年月日

R2 年 6 月 23 日 5 時限 ～ 7 時限

天候 気温 [°C] 湿度 [%]

R2 年 6 月 30 日 5 時限 ～ 7 時限

天候 気温 [°C] 湿度 [%]

実験レポート提出者

電子情報工学科 第 3 学年 8 番

氏名 織田 祐斗

提出年月日 年 月 日 提出

課題 1

課題内容

- (1) emacs などのテキストエディタを起動して showdata を編集する
- (2) 最初の行に, 実行に用いるシェルを以下の様にパスで追加する

```
#!/bin/bash  
hostname  
whoami  
pwd
```

- (3) エディタを保存して終了する.
- (4) 現在の showdata の許可条件を”ls -l”コマンドで確認する.
- (5) シェルスクリプトを実行可能とするための許可条件を設定する.
- (6) シェルスクリプトを実行する.

作成したスクリプト

```
#!/bin/bash  
hostname  
whoami  
pwd
```

結果

```
i1811408@LAPTOP-OQLL7GE0:~$ ls -l
total 675024
-rw-rw-rw- 1 i1811408 i1811408 685906562 Apr  5  2019 Anaconda3-2019.03-Linux-
drwxrwxrwx 1 i1811408 i1811408    4096 Jun 23 15:16 ShellKadai1
-rwxrwxrwx 1 i1811408 i1811408    3729 Apr 28 13:39 basic.sql
-rwxrwxrwx 1 i1811408 i1811408    8304 Nov 19  2018 bunsuu
-rw-rw-rw- 1 i1811408 i1811408     240 Nov 19  2018 bunsuu.c
-rw-rw-rw- 1 i1811408 i1811408     240 Nov 19  2018 bunsuu.c~
-rwxrwxrwx 1 i1811408 i1811408    8344 Nov  9  2018 coin
-rw-rw-rw- 1 i1811408 i1811408     694 Nov  9  2018 coin.c
-rw-rw-rw- 1 i1811408 i1811408     694 Nov  9  2018 coin.c~
lrwxrwxrwx 1 i1811408 i1811408      37 Apr 10  2019 cprogram -> /mnt/c/Users/
gram
drwxr-xr-x 1 i1811408 i1811408    4096 Dec 19  2019 eggx-0.93r5
-rw-rw-rw- 1 i1811408 i1811408 5222643 Sep  5  2015 eggx-0.93r5.tar.gz
-rwxrwxrwx 1 i1811408 i1811408    8304 Nov  2  2018 hello
-rw-rw-rw- 1 i1811408 i1811408      79 Nov  2  2018 hello.c
-rwxrwxrwx 1 i1811408 i1811408    8304 Nov 19  2018 hissan
-rw-rw-rw- 1 i1811408 i1811408     177 Nov 19  2018 hissan.c
-rw-rw-rw- 1 i1811408 i1811408     304 Nov 19  2018 hissan.c~
drwxr-xr-x 1 i1811408 i1811408    4096 Jun  4  2007 kadai2
drwxrwxrwx 1 i1811408 i1811408    4096 Jun  9 14:36 kadai4
-rw-rw-rw- 1 i1811408 i1811408     895 May 20  2019 kadai5.txt
-rw-rw-rw- 1 i1811408 i1811408    4503 May  9  2019 my_unix_kadai2
-rw-rw-rw- 1 i1811408 i1811408   4472 Jun  9 14:05 my_unix_kadai2.tgz
-rwxr-w-rw- 1 i1811408 i1811408      32 Jun 23 13:47 showdata
-rw-rw-rw- 1 i1811408 i1811408      20 Jun 23 13:45 showdata~
```

```
i1811408@LAPTOP-OQLL7GE0:~$ chmod u+x showdata
i1811408@LAPTOP-OQLL7GE0:~$ ./showdata
LAPTOP-OQLL7GE0
i1811408
/home/i1811408
```

考察

hostname は PC 名、whoami はユーザ名、pwd はカレントディレクトリの場所をそれぞれ表示するコマンドであるため、実行結果は正しいことがわかる。

課題2

課題内容

ShellKadai1 ディレクトリを作成して、ShellKadai1 ディレクトリに移動し、以下の仕様に沿ったシェルスクリプトを作成せよ。なお、ファイル名は Kadai2 とする。

仕様

- (1) カレントディレクトリに test という名前のディレクトリを作成する
- (2) test ディレクトリに移動する。
- (3) 現在のディレクトリを pwd コマンドで表示する。
- (4) 現在のディレクトリを ls コマンドで表示する。
- (5) echo コマンドを用いて“This is test”という文字列を、test.txt というファイルに書き込む。
- (6) 現在のディレクトリを ls コマンドで表示する。
- (7) test.txt の内容を cat コマンドで行番号つきで表示する。
- (8) ShellKadai1 ディレクトリに移動する
- (9) test ディレクトリ以下を削除する。

作成したスクリプト

```
mkdir test
cd test
pwd
ls -l
echo "This is test" > test.txt
ls -l
cat -n test.txt
cd ../
rm -r test
ls -l
```

結果

```
i1811408@LAPTOP-OQL7GEO:~/ShellKadai1$ ./Kadai2
/home/i1811408/ShellKadai1/test
total 0
total 0
-rw-rw-rw- 1 i1811408 i1811408 13 Jun 23 15:34 test.txt
      1 This is test
total 0
-r-xrw-rw- 1 i1811408 i1811408 105 Jun 23 14:08 Kadai2
-rwxrw-rw- 1 i1811408 i1811408  99 Jun 23 14:07 Kadai2~
```

考察

1 行目で test ディレクトリを作成し、移動できたことが確認できる。また、2 行目、3 行目から test ディレクトリにファイルやディレクトリが格納されていないこともわかる。4, 5 行目より、"This is test"の文が格納された test.txt が test ディレクトリ内に作成されたことが確認できる。7, 8 行目より、ShellKadai1 に移動したこと、test ディレクトリが削除されたことが確認できる。

よって、正しくシェルスクリプトが作成できたことが確認できる。

課題 3

課題内容

- (1) コマンド `yes` を適当な文字列を引数として実行せよ.
- (2) 実行されているコマンド `yes` のジョブを, サスペンド・ジョブにせよ.
- (3) 現在, 停止状態にあるジョブを表示させるコマンド `jobs` がある. `jobs` コマンドを実行して, サスペンド・ジョブを表示させよ.
- (4) ジョブを再開せよ.
- (5) ジョブを強制終了せよ.

結果

```
i1811408@LAPTOP-OQLL7GE0:~/ShellKadai1$ jobs
[1]+  Stopped                  yes job
i1811408@LAPTOP-OQLL7GE0:~/ShellKadai1$ fg
```

```
i1811408@LAPTOP-OQLL7GE0:~/ShellKadai1$ jobs
[1]+  Stopped                  yes job
i1811408@LAPTOP-OQLL7GE0:~/ShellKadai1$ kill %1
[1]+  Terminated              yes job
i1811408@LAPTOP-OQLL7GE0:~/ShellKadai1$ jobs
i1811408@LAPTOP-OQLL7GE0:~/ShellKadai1$
```

考察

それぞれのスクリーンショットの直前に `job` が連続して表示された。ジョブが停止していること、`kill` コマンドの後にジョブが存在しないことから、正しい実行結果を得られたことが確認できる。

課題 4

課題内容

- (1) コマンド `yes` を用いて、異なる文字列 A,B,C,D,E を引数として、5 個のバックグラウンド・ジョブとして実行するシェルスクリプトを作成せよ。
- (2) 作成したシェルスクリプトを実行せよ。
- (3) もう一方の端末ウィンドウでコマンド `ps` を用いて、コマンド `yes` のジョブ（プロセス）を確認せよ。
- (4) A, C, D を表示しているジョブを強制終了せよ。
- (5) 現在のジョブのプロセスを確認せよ。
- (6) 全てのコマンド `yes` のジョブを、それぞれ強制終了させよ。

作成したスクリプト

```
#!/bin/bash
yes A &
yes B &
yes C &
yes D &
yes E &
```

結果

```
i1811408@LAPTOP-QQL7GE0:~$ ps -a
PID TTY          TIME CMD
  6  tty1         00:00:00 bash
 52  tty1         00:00:00 emacs
 54  tty1         00:00:00 emacs
 56  tty2         00:00:00 bash
 69  tty1         00:00:00 yes
 70  tty1         00:00:00 yes
 71  tty1         00:00:00 yes
 72  tty1         00:00:00 yes
 73  tty1         00:00:00 yes
 74  tty2         00:00:00 ps
i1811408@LAPTOP-QQL7GE0:~$ kill 69 71 72
i1811408@LAPTOP-QQL7GE0:~$ ps -a
PID TTY          TIME CMD
  6  tty1         00:00:00 bash
 52  tty1         00:00:00 emacs
 54  tty1         00:00:00 emacs
 56  tty2         00:00:00 bash
 70  tty1         00:00:01 yes
 73  tty1         00:00:01 yes
 75  tty2         00:00:00 ps
i1811408@LAPTOP-QQL7GE0:~$ kill 70 73
i1811408@LAPTOP-QQL7GE0:~$ ps -a
PID TTY          TIME CMD
  6  tty1         00:00:00 bash
 52  tty1         00:00:00 emacs
 54  tty1         00:00:00 emacs
 56  tty2         00:00:00 bash
 76  tty2         00:00:00 ps
i1811408@LAPTOP-QQL7GE0:~$ s
```


考察

1 度目の確認で yes コマンドが 5 個、kill コマンドを入力した後の確認で yes コマンドが 2 個、再度 kill コマンドを入力した後の確認で yes コマンドが 0 個実行していることが確認できる。また、スクリプト実行後、AAAA……→BBBB……→CCCC……→DDDD……→EEEE……→AAAA……の順で表示され、kill コマンド実行後、BBBB……→EEEE……→BBBB……と表示されるようになり、再度 kill コマンド実行後は表示されなくなった。そのため、このスクリプトを正しく実行できたことが確認できる。

課題 5

課題内容

以下の注意に沿って，オリジナルのシェルスクリプトを作成せよ．

- ・ 3 つ以上のコマンドから構成されるものとする
- ・ 内容は自由とする
- ・ 他人とアイデアが重複しないように注意すること

作成したスクリプト

```
#!/bin/bash
~/cprogram/E_kadai1
~/cprogram/E_kadai2
~/cprogram/E_kadai3
~/cprogram/E_kadai4
```

結果

```
          整数型          実数型
1!=          1          1.0
2!=          2          2.0
3!=          6          6.0
4!=         24          24.0
5!=        120         120.0
6!=        720         720.0
7!=       5040        5040.0
8!=      40320       40320.0
9!=     362880      362880.0
10!=    3628800     3628800.0
11!=   39916800     39916800.0
12!=  479001600     479001600.0
13!= 1932053504     6227020800.0
14!= 1278945280     87178291200.0
15!= 2004310016     1307674368000.0
16!= 2004189184     20922789888000.0
17!= -288522240     355687428096000.0
18!= -898433024     6402373705728000.0
19!= 109641728      121645100408832000.0
20!= -2102132736    2432902008176640000.0
倍精度実数での桁数は53、浮動小数点数は2.220446e-16です
n=10
方法 1 : 2.7182818011463845131e+00
方法 2 : 2.7182818011463845131e+00
方法 3 : 2.7182818011463845131e+00
a=1.0
b=10.0
c=1.0
方法 1 : -0.101021,-9.898979
方法 2 : -9.898979,-0.101021
```

考察

前々回の実験で作成したプログラムを一気に実行するスクリプトを作成した。理想の実行結果が得られたので、正しくスクリプトを作成できたことが確認できる。