NPA2023 Lab Week02

Bash Shell Scripting

HelloWorld

1. Check all available shells

```
$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
$
```

2. Know where is your `bash`

```
$ which bash
/usr/bin/bash
```

3. Open an editor

```
$ nano hello.sh
```

4. Create a variable and print variable in `hello.sh`. The first line is called `Shebang`. All comments start with `#`

```
#!/usr/bin/bash
# declare a variable (no space between =)
STR="Hello"
# print variable
echo $STR
```

5. Change file permission

```
$ chmod +x hello.sh
```

6. Run bash shell script

```
$ ./hello.sh
Hello
$
```

Read Input from the Keyboard

`hello.sh`

```
#!/usr/bin/bash
STR="Hello"
read NAME
echo "$STR, $NAME"
```

```
$ ./hello.sh
Chotipat P.
Hello, Chotipat P.
$
```

Variables

The name of a variable can contain only letters (a to z or A to Z), numbers (0 to 9) or the underscore character (_). But numbers cannot be the first character. By convention, Unix shell variables will have their names in UPPERCASE.

```
# Scalar variables: can hold only one value at a time
NAME="Chotipat"
CARS=10

# To access the value stored in a variable, prefix its name with the dollar sign ($)
echo $NAME
echo $CARS
```

Special Variables

Special Variable	Description
$0	The filename of the current script.
$n	n is a positive decimal number corresponding to the position of an argument (the first argument is $1, the second argument is $2, and so on).
$#	The number of arguments supplied to a script.
$*	All the arguments are double quoted.
$@	All the arguments are individually double quoted.
$?	The exit status of the last command executed.
$$	The process number of the current shell. This is the process ID under which shell scripts are executing.
$!	The process number of the last background command.

`special.sh`

```
echo "File name of the current script: $0"
echo "First parameter: $1"
echo "Second parameter: $2"
echo "Third parameter: $3"
echo "\$*: $*"
echo "\$@: $@"
echo "Exit status of the last command": $?
echo "The process number of the current shell script: $$"
echo "The process number of the last background command: $!"
```

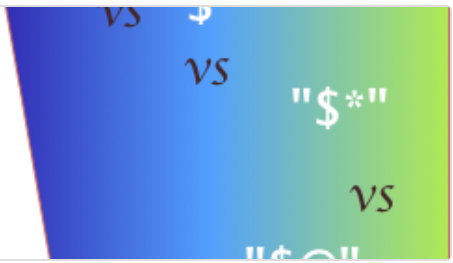
```
$ ./special.sh a "b c" d &
[1] 97
File name of the current script: ./special.sh
First parameter: a
Second parameter: b c
Third parameter: d
$*: a b c d
$@: a b c d
Exit status of the last command: 0
The process number of the current shell script: 97
The process number of the last background command:

[1]+  Done                    ./special.sh a "b c" d
$
$ ./special.sh a "b c" d
File name of the current script: ./special.sh
First parameter: a
Second parameter: b c
Third parameter: d
$*: a b c d
$@: a b c d
Exit status of the last command: 0
The process number of the current shell script: 98
The process number of the last background command: 97
$
$ ./special.sh a "b c" d
File name of the current script: ./special.sh
First parameter: a
Second parameter: b c
Third parameter: d
$*: a b c d
$@: a b c d
Exit status of the last command: 0
The process number of the current shell script: 99
The process number of the last background command: 97
$
```

What is the difference between $@ and $* ?

I think it is easier to understand this issue , in the following manner . Say you have this bash script :

https://mohamad-wael.medium.com/what-is-the-difference-between-and-66b0df1655fb



Decision Making

The if…else statements

- `if...fi`
- `if...else...fi`
- `if...elif...else...fi`

`decision1.sh`

```
!/usr/bin/bash

if [ $1 == $2 ]
then
   echo "Equal"
else
   echo "Not equal"
fi
```

```
$ ./decision1.sh 1 1
Equal
$ ./decision1.sh 1 2
Not equal
$
```

`decision2.sh`

```
#!/usr/bin/bash

if [ $1 == $2 ]
then
   echo "Equal"
elif [ $1 -gt $2 ]
then
```

```
   echo "first > second"
else
   echo "second > first"
fi
```

```
$ ./decision2.sh 1 1
Equal
$ ./decision2.sh 2 1
first > second
$ ./decision2.sh 1 2
second > first
$
```

Repetition

`for` loop and `while` loop

for Loop

`for1.sh`

```
#!/usr/bin/bash

for command in clear date ls
do
    sleep 1
    $command
done
```

```
$ ./for1.sh
Tue Dec  5 17:51:10 UTC 2023
array.sh      decision2.sh  for1.sh   special-star-vs-at.sh  variables.sh
decision1.sh  expr.sh     hello.sh  special.sh
$
```

`for2.sh`

```
#!/usr/bin/bash

for var in {1..10}
do
```

```
  echo $var
done
```

```
$ ./for2.sh
1
2
3
4
5
6
7
8
9
10
$
```

`for3.sh`

```
#!/usr/bin/bash

max=10
for  ((i=1; i<=max;  i++))
do
echo -n "$i "    # one case with echo without -n option
done
echo
```

```
$ ./for3.sh
1 2 3 4 5 6 7 8 9 10
$
```

while Loop

`while1.sh`

```
#!/usr/bin/bash

declare -i x
x=0
while [ $x -le 10 ]
do
  echo $x
```

```
  x=$((x+1))
done
```

```
$ ./while1.sh
0
1
2
3
4
5
6
7
8
9
10
$
```

`while2.sh`

```
#!/usr/bin/bash

i=1
while true
do
   sleep 1
   echo $i
   i=$((i+1))
done
```

```
# ./while2.sh
1
2
3
4
^C
$
```

Awk

Examples

```
$ speedtest-cli --list > servers.txt
$ cat servers.txt
Retrieving speedtest.net configuration...
20220) DTAC (Pathum Wan, Thailand) [27.22 km]
27203) ByteArk Co., Ltd. (Bangrak, Thailand) [27.28 km]
56644) Thailand Internet Exchange Point (Bangkok, Thailand) [30.46 km]
17560) TrueMove H (Bangkok, Thailand) [32.34 km]
23295) 3BB (Pathum Thani, Thailand) [43.82 km]
37242) SCM Technologies (Chonburi, Thailand) [45.10 km]
29053) 3BB (Samut Sakhon, Thailand) [57.71 km]
37240) SCM Technologies (Ayutthaya, Thailand) [75.70 km]
47818) MimoTech (Nakhon Pathom, Thailand) [81.02 km]
48164) AWN (Nakhon Pathom, Thailand) [81.02 km]
$
```

Awk print

```
$ awk '{print}' servers.txt
Retrieving speedtest.net configuration...
20220) DTAC (Pathum Wan, Thailand) [27.22 km]
27203) ByteArk Co., Ltd. (Bangrak, Thailand) [27.28 km]
56644) Thailand Internet Exchange Point (Bangkok, Thailand) [30.46 km]
17560) TrueMove H (Bangkok, Thailand) [32.34 km]
23295) 3BB (Pathum Thani, Thailand) [43.82 km]
37242) SCM Technologies (Chonburi, Thailand) [45.10 km]
29053) 3BB (Samut Sakhon, Thailand) [57.71 km]
37240) SCM Technologies (Ayutthaya, Thailand) [75.70 km]
47818) MimoTech (Nakhon Pathom, Thailand) [81.02 km]
48164) AWN (Nakhon Pathom, Thailand) [81.02 km]
$
```

Note that `'{print}'` and `'{print $0}'` are the same.

```
$ awk '{print $1}' servers.txt
Retrieving
20220)
27203)
56644)
17560)
23295)
37242)
29053)
37240)
47818)
```

```
48164)
$
```

```
$ awk '{print $2}' servers.txt
speedtest.net
DTAC
ByteArk
Thailand
TrueMove
3BB
SCM
3BB
SCM
MimoTech
AWN
$
```

We can use pipe |

```
$ speedtest-cli --list | awk '{print $1}'
Retrieving
20220)
27203)
56644)
17560)
23295)
37242)
29053)
37240)
47818)
48164)
$
```

Field separator is space by default. Change with `-F`

```
$ speedtest-cli --list | awk -F ')' '{print $1}'
Retrieving speedtest.net configuration...
16063
33968
47115
 7115
57015
16128
48165
```

```
29053
48164
42134
$
```

Add a pattern NR (Number of Records) > 1 to skip the first line (record).

```
$ speedtest-cli --list | awk -F ')' 'NR > 1 {print $1}'
23312
36978
 9830
27241
 7115
11823
22667
37033
16062
17691
$
```

Task 5

A Sample Solution

```
$ sudo apt install speedtest-cli
$ cat speedtest-lab.sh
#!/bin/bash

header=$(speedtest-cli --secure --csv-header)
echo $header > speedtest-chotipat.csv

servers=$(speedtest-cli --list | awk -F ')' 'NR > 1 {print $1}')
for server in $servers
do
    echo $(date) "Server ID: $server"
    speedtest-cli --secure --server $server --csv >> speedtest-chotipat.csv
    echo "Done"
done
$
```