

# Shell Scripts

- For each shell scripts in the Lecture slides,
  - Create each shell script (test1.sh to test27.sh)
  - Run each shell script
  - Take a snapshot of a shell script code and its running outputs
  - Also, explain what each shell script does

# test19.sh

```
babyearn@babyearn:~/Desktop
GNU nano 7.2
#!/bin/bash
echo "Enter directory name"
read newdir
`mkdir $newdir`
```

```
babyearn@babyearn:~$ nano test19.sh
babyearn@babyearn:~$ more test19.sh
#!/bin/bash
echo "Enter directory name"
read newdir
`mkdir $newdir` _____
babyearn@babyearn:~$ chmod 755 test19.sh
babyearn@babyearn:~$ ./test19.sh
Enter directory name
abcd _____
babyearn@babyearn:~$ ls
abcd email-verification Pictures Templates test12.sh test16.sh test1.sh test5.sh test9.sh
Desktop ict Public test test13.sh test17.sh test2.sh test6.sh Videos
Documents loop1.sh snap test10.sh test14.sh test18.sh test3.sh test7.sh
Downloads Music sum.sh test11.sh test15.sh test19.sh test4.sh test8.sh
```

สร้าง directory ใหม่ ด้วย mkdir ผ่าน shell

ใช้ชื่อ directory ที่อย่างต่อ

สร้างสำเร็จ

## test20.sh

```
babyearn@babyearn:~/Desktop
GNU nano 7.2
#!/bin/bash
echo "Enter directory name"
read ndir
if [ -d "$ndir" ]
then
echo "Directory exist"
else
`mkdir $ndir`
echo "Directory created"
fi
```

```
babyearn@babyearn:~$ nano test20.sh
babyearn@babyearn:~$ more test20.sh
#!/bin/bash
echo "Enter directory name"
read ndir
if [ -d "$ndir" ]
then
echo "Directory exist"
else
`mkdir $ndir`
echo "Directory created"
fi
babyearn@babyearn:~$ chmod 755 test19.sh
babyearn@babyearn:~$ chmod 755 test20.sh
babyearn@babyearn:~$ ./test20.sh
Enter directory name
defg
Directory created
babyearn@babyearn:~$ ./test20.sh
Enter directory name
abcd
Directory exist
babyearn@babyearn:~$
```

เก็บ directory → ตรวจสอบว่ามี directory ชื่อ

ถ้ามีแล้ว

ถ้าไม่มี

defg บันทึกผล "Directory created"

abcd มีแล้ว ผล "Directory exist"

# test21.sh

```
GNU nano 7.2
#!/bin/bash
file='book.txt'
while read line; do
echo $line
done < $file
```

test21.sh \*

```
GNU nano 7.2
1. Pro AngularJS
2. Learning JQquery
3. PHP Programming
4. CodeIgniter 3
```

book.txt

```
babyearn@babyearn:~$ nano test21.sh
babyearn@babyearn:~$ nano book.txt
babyearn@babyearn:~$ more test21.sh
#!/bin/bash
file='book.txt'
while read line; do
echo $line
done < $file
babyearn@babyearn:~$ chmod 755 test21.sh
babyearn@babyearn:~$ ./test21.sh
1. Pro AngularJS
2. Learning JQquery
3. PHP Programming
4. CodeIgniter 3
babyearn@babyearn:~$
```

ก່າວນອື່ດ້ວຍ file ຖໍ່ມະດີ

ດ້ານຂອງ: ໜັດຕັບ

## test22.sh

```
GNU nano 7.2                                         test22.sh
#!/bin/bash
echo "Enter filename to remove"
read fn
rm -i $fn
```

```
babyearn@babyearn:~$ nano test22.sh
babyearn@babyearn:~$ more test22.sh
#!/bin/bash
echo "Enter filename to remove"
read fn
rm -i $fn
L
babyearn@babyearn:~$ ls
abcd      Desktop      ict      Public      test      test13.sh  test17.sh  test20.sh  test3.sh  test7.sh
book.txt   Documents    loop1.sh snap      test10.sh  test14.sh  test18.sh  test21.sh  test4.sh  test8.sh
defg      Downloads    Music     sum.sh      test11.sh  test15.sh  test19.sh  test22.sh  test5.sh  test9.sh
delete_file.sh email-verification Pictures  Templates test12.sh  test16.sh  test1.sh   test2.sh   test6.sh  Videos
babyearn@babyearn:~$ chmod 755 test22.sh
babyearn@babyearn:~$ bash test22.sh
Enter filename to remove
delete_file.sh
rm: remove regular file 'delete_file.sh'? y
babyearn@babyearn:~$ ls
abcd      Documents    loop1.sh snap      test10.sh  test14.sh  test18.sh  test21.sh  test4.sh  test8.sh
book.txt  Downloads    Music     sum.sh      test11.sh  test15.sh  test19.sh  test22.sh  test5.sh  test9.sh
defg      email-verification Pictures  Templates test12.sh  test16.sh  test1.sh   test2.sh   test6.sh  Videos
Desktop   ict          Public    test      test13.sh  test17.sh  test20.sh  test3.sh  test7.sh
babyearn@babyearn:~$
```

↑ รับชื่อ file ที่ต้องการลบ  
 ↑ อ่าน file  
 ↑ ลบ file

↑ กด y เพื่อยืนยันการลบ

## test23.sh

```
babyearn@babyearn:~$ nano test23.sh
babyearn@babyearn:~$ more test23.sh
#!/bin/bash
echo "Before appending the file"
cat book.txt
echo "Learning Laravel 5">>> book.txt
echo "After appending the file"
cat book.txt
babyearn@babyearn:~$ chmod 755 test23.sh
babyearn@babyearn:~$ bash test23.sh
Before appending the file
1. Pro AngularJS
2. Learning JQquery
3. PHP Programming
4. CodeIgniter 3
Learning Laravel 5
After appending the file
1. Pro AngularJS
2. Learning JQquery
3. PHP Programming
4. CodeIgniter 3
Learning Laravel 5
Learning Laravel 5
babyearn@babyearn:~$
```

```
GNU nano 7.2
#!/bin/bash
echo "Before appending the file"
cat book.txt
echo "Learning Laravel 5">>> book.txt
echo "After appending the file"
```

→ ॥สูตรเน้นนำก่อนเพิ่ม  
→ เพิ่ม "Learning Laravel 5" ที่ท้าย file  
→ ॥สูตรเน้นนำหลังเพิ่ม

## test24.sh

```
GNU nano 7.2
#!/bin/bash
filename=$1
if [ -f "$filename" ]; then
echo "File exists"
else
echo "File does not exist"
fi
```

## test24.sh

```
babyearn@babyearn:~$ nano test24.sh
babyearn@babyearn:~$ more test24.sh
#!/bin/bash
filename=$1
if [ -f "$filename" ]; then
echo "File exists"
else
echo "File does not exist"
fi
```

ตรวจสอบว่า file นี้เป็น  
ถ้ามี  
ถ้าไม่มี

```
babyearn@babyearn:~$ chmod 755 test24.sh
babyearn@babyearn:~$ bash test24.sh book.txt
File exists
babyearn@babyearn:~$ bash test24.sh book1.txt
File does not exist
babyearn@babyearn:~$
```

## test25.sh

```
babyearn@babyearn:~$ nano test25.sh  
babyearn@babyearn:~$ more test25.sh
```

```
#!/bin/bash  
Year=`date +%Y`  
Month=`date +%m`  
Day=`date +%d`  
Hour=`date +%H`  
Minute=`date +%M`  
Second=`date +%S`  
echo `date`  
echo "Current Date is: $Day-$Month-$Year"  
echo "Current Time is: $Hour:$Minute:$Second"
```

```
GNU nano 7.2  
#!/bin/bash  
Year=`date +%Y`  
Month=`date +%m`  
Day=`date +%d`  
Hour=`date +%H`  
Minute=`date +%M`  
Second=`date +%S`  
echo `date`  
echo "Current Date is: $Day-$Month-$Year"  
echo "Current Time is: $Hour:$Minute:$Second"
```

ກ່າວນິຕັ້ງແປງ

॥ສອງວັນທີ॥ລະເກລາປ່ຽນຸ້ນ້າ

## test26.sh

```
GNU nano 7.2
#!/bin/bash
echo "Wait command" &
process_id=$!
wait $process_id
echo "Exited with status $?"
```

test26.sh

```
babyearn@babyearn:~$ nano test26.sh
babyearn@babyearn:~$ more test26.sh
#!/bin/bash
echo "Wait command" &
process_id=$! _____ → เก็บ PID
wait $process_id _____ → รับ process ที่งานสำเร็จ
echo "Exited with status $?" _____ → เมื่อเสร็จแล้วจะแสดง "Exited with status $?"
babyearn@babyearn:~$ chmod 755 test26.sh
babyearn@babyearn:~$ ./test26.sh
Wait command
Exited with status 0 _____ → 0 = process สำเร็จ
babyearn@babyearn:~$
```

→ เก็บ PID  
→ รับ process ที่งานสำเร็จ  
→ เมื่อเสร็จแล้วจะแสดง "Exited with status \$?"  
0 = process สำเร็จ

## test27.sh

```
GNU nano 7.2
#!/bin/bash
echo "Wait for 5 seconds"
sleep 5
echo "Completed"
```

test27.sh \*

```
babyearn@babyearn:~$ nano test27.sh
babyearn@babyearn:~$ more test27.sh
#!/bin/bash
echo "Wait for 5 seconds" _____ → แสดงข้อความในรูป 5 วินาที
sleep 5 _____ → หยุดการทำงาน 5 วินาที
echo "Completed" _____
babyearn@babyearn:~$ chmod 755 test27.sh
babyearn@babyearn:~$ ./test27.sh
"Wait for 5 seconds"
"Completed"
babyearn@babyearn:~$
```

แสดงข้อความในรูป 5 วินาที

หยุดการทำงาน 5 วินาที

หลัง 5 วินาที จะแสดง "Completed"

# Environment Variables

- Look at the content of “.profile”, “.bashrc”
- Set “ignoreeof” and “noclobber”
- Set new prompt
- Set new color of the prompt

- Look at the content of “.profile”, “.bashrc”

Thankamol Saekeang Sec:1 6687023

```
babyearn@babyearn:~$ cat ~/.profile
# ~/.profile: executed by the command interpreter for login shells.
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
# exists.
# see /usr/share/doc/bash/examples/startup-files for examples.
# the files are located in the bash-doc package.

# the default umask is set in /etc/profile; for setting the umask
# for ssh logins, install and configure the libpam-umask package.
#umask 022

# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
fi
```

```
babyearn@babyearn:~$ cat ~/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "**" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus is in a terminal window
# should be on the output of commands, not on the prompt
#force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >& /dev/null; then
        # We have color support; assume it's compliant with Ecma-48
        # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
        # a case would tend to support setf rather than setaf.)
        color_prompt=yes
    else
        color_prompt=
    fi
fi

if [ "$color_prompt" = yes ]; then
    PS1='${debian_chroot:+($debian_chroot)}[\u033[01;32m]\u0@\h[\u033[00m]:[\u033[01;34m]\w[\u033[00m]\$ '
else
    PS1='${debian_chroot:+($debian_chroot)}\u0@\h:\w\$ '
fi
unset color_prompt force_color_prompt

# If this is an xterm set the title to user@host:dir
case "TERM" in
xterm*|rxvt*)
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u0@\h: \w\]\$PS1"
    ;;
*)
    ;;
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    alias dir='dir --color=auto'
    alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
# alias alert='notify-send --urgency=low -i "$(($? == 0)) && echo terminal || echo error" "$(history|tail -n1|sed -e '\''s/^s*\([0-9]\)\+\s*/\1/g')"

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ./bash_aliases ]; then
    . ./bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
babyearn@babyearn:~$
```

- Set “ignoreeof” and “noclobber”
- Set new prompt
- Set new color of the prompt

```
babyearn@babyearn:~$ set -o ignoreeof
babyearn@babyearn:~$ set -o noclobber
babyearn@babyearn:~$ PS1='[\u0@h \W]\$ '
[babyearn@babyearn ~]$ PS1='\[\e[1;35m\][\u0@h \W]\$\[\e[0m\] '
[babyearn@babyearn ~]$ 
```

# Programming

- Install C compiler (gcc – GNU C Compiler)
- Create first.c program
- Compile first.c program
- Run first.c program
- Take a snapshot of first.c running and its output

# Programming

```
babyearn@babyearn:~/Desktop$ sudo apt update
[sudo] password for babyearn:
Ign:1 http://ports.ubuntu.com/ubuntu-ports noble InRelease
Ign:2 http://ports.ubuntu.com/ubuntu-ports noble-updates InRelease
Ign:3 http://ports.ubuntu.com/ubuntu-ports noble-backports InRelease
Ign:4 http://ports.ubuntu.com/ubuntu-ports noble-security InRelease
Ign:1 http://ports.ubuntu.com/ubuntu-ports noble InRelease
Ign:2 http://ports.ubuntu.com/ubuntu-ports noble-updates InRelease
Ign:3 http://ports.ubuntu.com/ubuntu-ports noble-backports InRelease
Ign:4 http://ports.ubuntu.com/ubuntu-ports noble-security InRelease
Ign:1 http://ports.ubuntu.com/ubuntu-ports noble InRelease
Ign:2 http://ports.ubuntu.com/ubuntu-ports noble-updates InRelease
Ign:3 http://ports.ubuntu.com/ubuntu-ports noble-backports InRelease
Ign:4 http://ports.ubuntu.com/ubuntu-ports noble-security InRelease
Err:1 http://ports.ubuntu.com/ubuntu-ports noble InRelease
  Connection failed [IP: 185.125.190.36 80]
Err:2 http://ports.ubuntu.com/ubuntu-ports noble-updates InRelease
  Connection failed [IP: 185.125.190.36 80]
Err:3 http://ports.ubuntu.com/ubuntu-ports noble-backports InRelease
  Connection failed [IP: 185.125.190.36 80]
Err:4 http://ports.ubuntu.com/ubuntu-ports noble-security InRelease
  Connection failed [IP: 185.125.190.36 80]
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
7 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: Failed to fetch http://ports.ubuntu.com/ubuntu-ports/dists/noble/InRelease  Connection failed [IP: 185.125.190.36 80]
W: Failed to fetch http://ports.ubuntu.com/ubuntu-ports/dists/noble-updates/InRelease  Connection failed [IP: 185.125.190.36 80]
W: Failed to fetch http://ports.ubuntu.com/ubuntu-ports/dists/noble-backports/InRelease  Connection failed [IP: 185.125.190.36 80]
W: Failed to fetch http://ports.ubuntu.com/ubuntu-ports/dists/noble-security/InRelease  Connection failed [IP: 185.125.190.36 80]
W: Some index files failed to download. They have been ignored, or old ones used instead.
babyearn@babyearn:~/Desktop$ cd
babyearn@babyearn:~$ sudo apt install gcc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:13.2.0-7ubuntu1).
gcc set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
babyearn@babyearn:~$ brew install gcc
Command 'brew' not found, did you mean:
 command 'brec' from deb bplay (0.991-10.2)
 command 'qbrew' from deb qbrew (0.4.1-8build1)
Try: sudo apt install <deb name>
babyearn@babyearn:~$ nano first.c
babyearn@babyearn:~$ gcc first.c -o first
babyearn@babyearn:~$ ./first
Hello, World!
```

- Install C compiler (gcc – GNU C Compiler)
- Create first.c program
- Compile first.c program
- Run first.c program
- Take a snapshot of first.c running and its output

```
babyearn@babyearn:~$ more first.c
#include <stdio.h>

void main
{
    int x;

    x = 10;
    printf(" Hello World \n");
    printf(" value of x is %d\n", x);
}
```

# Programming

- Create, compile and run fork1.c program
- Create, compile and run fork2.c program
- Create, compile and run fork3.c program

# • Create, compile and run fork1.c program

```

babyearn@babyearn:~$ nano fork1.c
babyearn@babyearn:~$ more fork1.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main (int argc, char* argv[])
{
    int pid, ppid;
    int x;

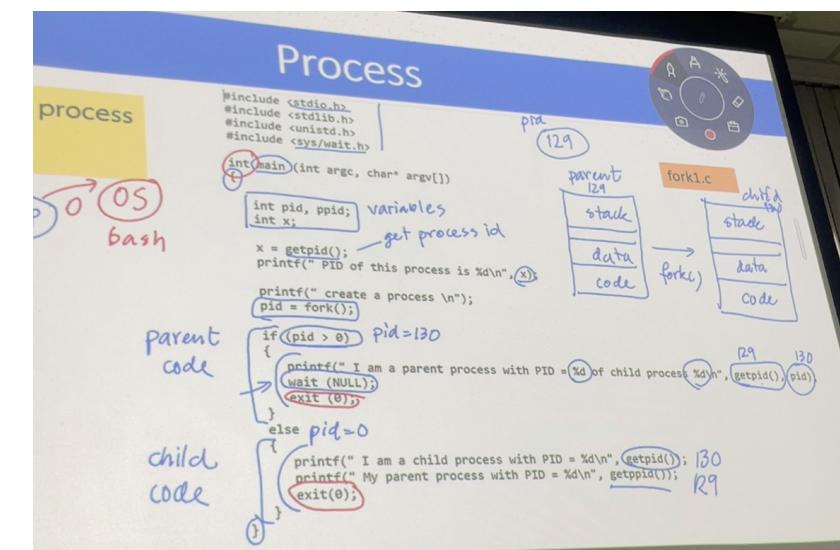
    x = getpid();
    printf(" PID of this process is %d\n", x);

    printf(" create a process \n");
    pid = fork();

    if (pid > 0)
    {
        printf(" I am a parent process with PID = %d of child process %d\n", getpid(), pid);
        wait(NULL);
        exit(0);
    }
    else
    {
        printf(" I am a child process with PID = %d\n", getpid());
        printf(" My parent process with PID = %d\n", getppid());
        exit(0);
    }
}

babyearn@babyearn:~$ gcc fork1.c -o fork1
babyearn@babyearn:~$ ./fork1
PID of this process is 5670
create a process
I am a parent process with PID = 5670 of child process 5671
I am a child process with PID = 5671
My parent process with PID = 5670
babyearn@babyearn:~$ 

```



- Create, compile and run fork2.c program

```
babyearn@babyearn:~$ nano fork2.c
babyearn@babyearn:~$ more fork2.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

void main() {
    int pid, ppid;

    printf("Parent process ID is %d\n", getpid());
    pid = fork();
    for (int i = 0; i < 2; i++) {
        pid = fork();
        printf("Process ID = %d, Parent ID = %d, Loop #%d\n", getpid(), getppid(), i);
    }

    printf("Process ID %d is done\n", getpid());
    wait(NULL);
    exit(0);
}

babyearn@babyearn:~$ gcc fork2.c -o fork2
babyearn@babyearn:~$ ./fork2
Parent process ID is 5701
Process ID = 5701, Parent ID = 4125, Loop #0
Process ID = 5703, Parent ID = 5701, Loop #0
Process ID = 5702, Parent ID = 5701, Loop #0
Process ID = 5701, Parent ID = 4125, Loop #1
Process ID = 5704, Parent ID = 5702, Loop #0
Process ID 5701 is done
Process ID = 5705, Parent ID = 5701, Loop #1
Process ID 5705 is done
Process ID = 5702, Parent ID = 5701, Loop #1
Process ID = 5703, Parent ID = 5701, Loop #1
Process ID 5702 is done
Process ID 5703 is done
Process ID = 5708, Parent ID = 5702, Loop #1
Process ID = 5706, Parent ID = 5703, Loop #1
Process ID 5708 is done
Process ID 5706 is done
Process ID = 5704, Parent ID = 5702, Loop #1
Process ID 5704 is done
Process ID = 5707, Parent ID = 5704, Loop #1
Process ID 5707 is done
babyearn@babyearn:~$
```

- Create, compile and run fork3.c program

```
babyearn@babyearn:~$ nano fork3.c
babyearn@babyearn:~$ more fork3.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
    printf("hello world (pid:%d)\n", (int) getpid());
    int rc = fork();
    if (rc < 0) {
        fprintf(stderr, "fork failed\n");
        exit(1);
    } else if (rc == 0) {
        printf("hello, I am child (pid:%d)\n", (int) getpid());
    } else {
        printf("hello, I am parent of %d (pid:%d)\n", rc, (int) getpid());
    }
    return 0;
}

babyearn@babyearn:~$ gcc fork3.c -o fork3
babyearn@babyearn:~$ ./fork3
hello world (pid:5731)
hello, I am parent of 5732 (pid:5731)
hello, I am child (pid:5732)
babyearn@babyearn:~$ █
```