



**សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ**

**Royal University of Phnom Penh**

**ដេប៉ាតឺម៉ង់៖ ព័ត៌មានវិទ្យា**

**មុខវិជ្ជា៖ មូលដ្ឋានគ្រឹះនៃ Linux System**

**បង្រៀនដោយ សាស្ត្រាចារ្យ បណ្ឌិត អ៊ុក ឃាន**

**ប្រធានបទ៖ ការគ្រប់គ្រងអ្នកប្រើប្រាស់ Linux System ជាមួយ Shell Script**

**អ្សបអ្សង នឹងស្រាវជ្រាវដោយក្រុមទី ៦ (ជំនាន់១៨)**

**សមាជិក**

ប៉េង ឆឹមឡេង

តោត តានិត្យា

ហេង ប៊ុនហាក់

កេវ ឈុនលី

ផង់ រតនៈ

អុន ប៊ុនធឿន

វ៉ែត សុយាង

**ឆ្នាំសិក្សា ២០១៧-២០១៨**

## មាតិកា

ចំណងជើងអត្ថបទ	ទំព័រ
១. សេចក្តីផ្តើម .....	1
២. គោលបំណង.....	1
៣. ទំហំការងារ.....	2
៤. ការអនុវត្ត.....	2
៥. សេចក្តីសន្និដ្ឋាន .....	7
៦. Reference.....	8

## 1. សេចក្តីផ្តើម

នៅក្នុងសកលលោកយើងនេះជាក់ស្តែងណាស់ប្រាកដជាមានការ access ដើម្បី ទាញយក ទិន្នន័យ រក្សាទុកទិន្នន័យ។ប្រសិនបើមិនមានការគ្រប់គ្រងត្រឹមត្រូវ ចំពោះការ access នោះទេ វាអាចបណ្តាលឲ្យមានបញ្ហា។ចំពោះការគ្រប់គ្រងសម្រាប់ការប្រើប្រាស់មាន OS(operating system) ជាច្រើនសម្រាប់ គ្រប់គ្រង។នៅក្នុងនោះមាន OS មួយដែលយើងនឹងធ្វើការលើកយក មកប្រើប្រាស់គឺ Cent OS6 ដែលជាប្រភេទ Open Source។Centos 6 ជាការបូកផ្សំគ្នាឡើង រវាង RED hat ជាមួយនិង fedora។

ដូចគ្នានេះដែរទើបមានការបណ្តុះបណ្តាលដោយ សាស្ត្រាចារ្យ បណ្ឌិត អ៊ុក យាន បានដាក់ នូវប្រធានបទ សម្រាប់គ្រប់គ្រង User។

## 2. គោលបំណង

ដូចដែលយើងបានរៀបរាប់ពីខាងលើនោះជាក់ស្តែងថា ការ គ្រប់គ្រងទៅលើការ access វា ពិតជាមានសារៈសំខាន់។ដូចនេះយើងក៏បានលើកយកនូវបញ្ហាមួយមកដោះស្រាយដូចខាង ក្រោម

Plan your CentOS Server to be a fileserver with 200 users who will be assigned 10GB space of your hard disk and write a shell script to create 200 users and control these users from a file called user.txt

ដូចនេះយើងនឹងធ្វើការ បង្កើត shell script និង ធ្វើការប៉ាន់ស្មានទំហំនិងតម្លៃ នៃ ម៉ាស៊ីន fileserver របស់យើង។

### 3. ទំហំការងារ

ចំពោះ fileserver អាចឱ្យ User access បាន 200 នាក់។ shell script អាចគ្រប់គ្រង User ដូចជា

- create new user
- update user password
- update user information
- create group
- add user to group

### 4. ការអនុវត្ត

#### 1. តម្រូវការចំពោះម៉ាស៊ីន Server

ដើម្បីបង្កើតបានជា File Server ដែលអនុញ្ញាតឱ្យអ្នកប្រើប្រាស់ចំនួន ២០០ នាក់ប្រើប្រាស់ ទាមទារឱ្យមាននូវ ម៉ាស៊ីន Server មួយដែលមានតម្រូវការដូចខាងក្រោម៖

- CPU Xeon processor E7-8800 v4/v3
- RAM 32GB
- Hard Disk 2.5TB
- ដំណើរការដោយ CentOS
- Samba File Server

តាមរយៈការគ្រោងទុកនេះ Server អាចមានតម្លៃចន្លោះពី \$1700 ទៅ \$2000 ។ ហើយដើម្បីងាយស្រួលក្នុងការគ្រប់គ្រង អ្នកប្រើប្រាស់ចំនួន ២០០ នាក់បានងាយស្រួលនោះ គឺគេចាំបាច់ត្រូវប្រើ Shell Script ដែលរៀបចំសរសេរទុកជាមុន។

#### 2. អំពី Shell Script

##### 2.1. Shell Script ជាអ្វី?

Shell Script គឺជាបណ្តុំនៃ Computer Command Line ដែរដំណើរការនៅលើ Shell ដែលត្រូវបានសរសេរចូលគ្នាទៅតាមលក្ខខណ្ឌ និងការចង់បាន នៅក្នុង File មួយដែលហៅថា

Script ។ ដើម្បីភាពងាយស្រួលរបស់អ្នកប្រើប្រាស់ Command Line ត្រូវបានគេសរសេរជាមួយ Options របស់វា ទៅក្នុង Script មួយ ។ ខាងក្រោមនេះគឺជា Script ដែលនឹងប្រើសម្រាប់គ្រប់គ្រងលើ អ្នកប្រើប្រាស់ និងប្រតិបត្តការមួយចំនួន ។

## 2.2. Command ដែលធ្វើការជាមួយអ្នកប្រើប្រាស់ ហើយប្រើនៅក្នុង shell script

- useradd សម្រាប់បន្ថែមអ្នកប្រើប្រាស់ថ្មី
- userdel សម្រាប់លុបអ្នកប្រើប្រាស់
- passwd សម្រាប់ប្តូរពាក្យសម្ងាត់របស់អ្នកប្រើប្រាស់
- usermod សម្រាប់កែប្រែព័ត៌មានរបស់អ្នកប្រើប្រាស់ (ការពណ៌នា ឈ្មោះសម្រាប់ login ជាដើម)
- chfn សម្រាប់កែប្រែព័ត៌មានរបស់អ្នកប្រើប្រាស់ (ឈ្មោះ ជាដើម)
- chage សម្រាប់បង្ហាញព័ត៌មានពីអាយុកាលរបស់ Password
- groupadd សម្រាប់បង្កើតក្រុមអ្នកប្រើប្រាស់
- groupdel សម្រាប់លុបក្រុមអ្នកប្រើប្រាស់

ដើម្បីងាយស្រួល ក្នុងការសរសេរ command ទាំងអស់ត្រូវបានរៀបចំជា function និងមានជា menu ងាយស្រួលសម្រាប់ការប្រើប្រាស់។ ខាងក្រោមនេះគឺជា shell script ទាំងមូល ។

## 2.3. Shell Script code

```
#!/bin/sh
# Group 6, Members:
# Peng Chhaileng
# Phang Ratanak
# Ret Sokheang
# Heng Bunhak
# On Bunthoeurn
# Kev Chhunly
# Phot Pheanitya

# Functions Area
check_root_user() {
    if [ "$EUID" -ne 0 ]
        then echo "[-] Permission denied! Are you root?"
        exit
    fi
}
```

```

menu() {
    echo "#-----MENU-----"
    echo "#      1. Create Users from File      "
    echo "#      2. Delete User                  "
    echo "#      3. Update User                  "
    echo "#      4. Create Group                  "
    echo "#      5. Delete group                  "
    echo "#      6. Add User to Group            "
    echo "#      7. Show all users in system      "
    echo "#      8. Show all groups in system     "
    echo "#      9. Show account information      "
    echo "#      0. Exit                          "
    echo -n "#      Choose options: "
}

menu_update_user() {
    echo "#-----MENU-----"
    echo "#      1. Update Information              "
    echo "#      2. Change Password                "
    echo "#      3. Add Description to User         "
    echo "#      4. Change Username (Login Name)    "
    echo "#      0. Back                            "
    echo -n "#      [Update User] options: "
}

pause() {
    echo -n $1
    read
}

add_user() {
    egrep "^$1" /etc/passwd >/dev/null
    if [ $? -eq 0 ]; then
        echo "[-] User $1 exists!"
    else
        pass=$(perl -e 'print crypt($ARGV[0], "password")' $2)
        useradd -m -p $pass $1
        [ $? -eq 0 ] && echo "[+] User has been added to system!" || echo
        "[-] Failed to add a user!"
    fi
}

controller() {
    users=()
    while IFS=' ' read -r user || [[ -n "$user" ]]; do
        if ! [ -z $user ]; then
            users+=("$user")
        fi
    done < "$1"
}

```

```

echo "[+] Found ${#users[@]} users"
while true
do
    menu
    read choose
    case $choose in
        "1") count=1
            for i in ${users[@]}
            do
                IFS=':' read -ra record <<< "$i"
                echo "User record : $count"
                echo "Username: ${record[0]}"
                add_user ${record[0]} ${record[1]}
                count=`expr $count + 1`
            done
            pause "Press enter to continue..."
        ;;
        "2") read -p "[*] Enter username: " usr
            userdel -r $usr > /dev/null 2>&1
            del_status=$?
            if [ $del_status -eq 0 ]; then
                echo "[+] User $usr deleted successfully!"
            else
                echo "[-] User $usr is not exist!"
            fi
            pause "Press enter to continue..."
        ;;
        "3") read -p "[*] Enter username: " usr
            id $usr > /dev/null 2>&1
            usr_status=$?
            if [ $usr_status -eq 0 ]; then
                while true
                do
                    menu_update_user
                    read c
                    case $c in
                        "1") chfn $usr
                            pause "Press enter to
continue..."
                        ;;
                        "2") passwd $usr
                            pause "Press enter to
continue..."
                        ;;
                        "3") read -p "[*] Description
(no space): " cmt
                            usermod -c $cmt $usr
                            cat /etc/passwd | grep $usr
                            pause "Press enter to
continue..."
                    ;;
                done
            fi
        ;;
    esac
done

```

```

                                "4")  read -p "[*] Enter new
username: " n_usr                                usermod -l $n_usr $usr
                                                echo "[+] Username update

successfully!"                                pause "Press enter to

continue..."                                ;;
                                                "0")  break
                                                ;;
                                esac
                                clear
                                done
                                else
                                echo "[-] User $usr is not exist!"
                                pause "Press enter to continue..."
                                fi
                                ;;
                                "4")  #clear
                                read -p "[*] Enter group name: " gro
                                groupadd $gro
                                pause "Press enter to continue..."
                                ;;
                                "5")  #clear
                                read -p "[*] Enter group name: " gro
                                groupdel $gro
                                pause "Press enter to continue..."
                                ;;
                                "6")  #clear
                                read -p "[*] Enter group name: " gro
                                read -p "[*] Enter user name: " usr
                                usermod -G $gro $usr
                                pause "Press enter to continue..."
                                ;;
                                "7")  clear
                                cut -d: -f1 /etc/passwd
                                pause "Press enter to continue..."
                                ;;
                                "8")  clear
                                cat /etc/group
                                pause "Press enter to continue..."
                                ;;
                                "9")  #clear
                                read -p "[*] Enter username: " usr
                                id $usr > /dev/null 2>&1
                                usr_status=$?
                                if [ $usr_status -eq 0 ]; then
                                    chage -l $usr
                                    pause "Press enter to continue..."
                                else
                                    echo "[-] User not found!"

```



```

                                pause "Press enter to continue..."
                                fi
                                ;;
                                "0") exit
                                ;;
                                *) echo "Command Not Found"
                                ;;
                                esac
                                clear
                                done
                                }

main() {
    read -p "[*] Enter users file name (users.txt): " file_name
    #file_name="users.txt"

    if [ -z $file_name ]; then
        file_name="users.txt"
    fi

    if ! [ -s $file_name ]; then
        echo "[-] File $file_name is not found!"
    else
        echo "[+] Using file $file_name..."
        echo "[+] Reading user from file..."

        controller $file_name
    fi
}

# Main Area
check_root_user
main

```

## 5.សេចក្តីសន្និដ្ឋាន

ក្រោយពីបានធ្វើការមង្កើត shell script រួចមកយើងពិតជាបានទទួលបាននូវការគ្រប់គ្រង user បានពិតមែន។ដោយយើងអាចធ្វើការបង្កើត លុប កែប្រែ ព័ត៌មាន ពាក្យសម្ងាត់បាន។ហើយមានតែ root ប៉ុន្មោះដែលអាចធ្វើការ កែប្រែបាន។

## 6. Reference

-សៀវភៅ មូលដ្ឋានគ្រឹះនៃប្រព័ន្ធ Linux system(CentOS 6) រៀបរៀងដោយ សាស្ត្រាចារ្យ  
បណ្ឌិត អ៊ុក ឃាន

-<http://www.chantracomputer.com/newpc.html>