

CAN THO UNIVERSITY
COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY
OPERATING SYSTEMS (CT104H)
LAB #2

Student name: **Bùi Tuấn Anh**

ID:B2405065

- **Submission:** Students submit 1 file named *StudentName_ID_CT104H_Lab02.pdf* to the Google classroom (where *StudentName* is the student's name, and *ID* is the student's ID).
- **Instructions on how to present in the report file**
 - For each question, students **MUST** provide the commands/scripts **AND** screenshots of the commands used and/or the content of files/scripts, **CLEARLY**. Note: *the screenshot needs to include the name of the Ubuntu Virtual machine*.
 - *Student creates an Ubuntu Virtual machine named UbuntuID (ID is the student's ID)*. For example, we have a virtual machine named **Ubuntu20043**, such that the student *ID* is **20043**.

Question 0: Navigate to your home directory

Answer: \$cd



Study from the [Linux CLI Fundamentals](#) (Sections: Nano Text Editor, Shell Scripting Basics), the book An Introduction to the Linux Command Shell for Beginners, and the book Linux Bible of Christopher Negus (Chapter 5: Working with the Text Files, and Chapter 7: Writing Simple Shell Scripts)

Solve all the questions from 1 to 10 in the Section Exercises of Chapter 5 (Linux Bible of Christopher Negus)

1. Copy the /etc/services fi le to the /tmp directory. Open the /tmp/services

fi le in vim, and search for the term WorldWideWeb. Change that to read World

Wide Web.

Answer:

```
~cp /etc/services /tmp/services
```

```
~cp vim /tmp/services
```

```
student@student:~$ cp /etc/services /tmp/services
student@student:~$ vim /tmp/services
```

/WorldWideWeb

:wq

```
gopher      70/tcp          # Internet Gopher
finger      79/tcp
http        80/tcp          www          # World Wide Web HTTP
link        87/tcp          ttylink
kerberos    88/tcp          kerberos5 krb5 kerberos-sec # Kerberos v5
kerberos    88/udp          kerberos5 krb5 kerberos-sec # Kerberos v5
supdup      95/tcp
hostnames   101/tcp         hostname      # usually from sri-nic
iso-tsap    102/tcp         tsap          # part of ISODE
acr-nema    104/tcp         dicom         # Digital Imag. & Comm. 300
acr-nema    104/udp         dicom
csnet-ns    105/tcp         cso-ns        # also used by CSO name server
csnet-ns    105/udp         cso-ns
rtelnet     107/tcp          # Remote Telnet
-- INSERT --                                     53,33-62      7%
```

2. Find the following paragraph in your /tmp/services file (if it is not there, choose

a different paragraph) , and move it to the end of that file.

Note that it is presently the policy of IANA to assign a single

well-known

port number for both TCP and UDP; hence, most entries here have two

entries

even if the protocol doesn't support UDP operations.

Updated from RFC 1700, "Assigned Numbers" (October 1994). Not all

ports

are included, only the more common ones.

Answer:

~\$ vim /tmp/services

Press shift + v to access visual line mod

Choose line to copy

Press shift + d to copy and cut

Press shift + g to go the end of that file

Press shift + d to paste

```
#=====
# The remaining port numbers are not as allocated by IANA.
#=====

# Kerberos (Project Athena/MIT) services
# Note that these are for Kerberos v4, and are unofficial.  Sites running
# v4 should uncomment these and comment out the v5 entries above.
#
# Local services
```

590,1 Bot

3. Using ex mode, search for every occurrence of the term tcp (case-sensitive) in

your /tmp/services file and change it to WHATEVER.

Answer:

~\$ vim /tmp/services

:%s/tcp/WHATEVER/g

```
sgi-cad      17004/WHATEVER      # Cluster Admin daemon
isdnlog      20011/WHATEVER      # isdn logging system
isdnlog      20011/udp
vboxd        20012/WHATEVER      # voice box system
vboxd        20012/udp
binkp        24554/WHATEVER      # binkp fidonet protocol
asp          27374/WHATEVER      # Address Search Protocol
asp          27374/udp
csync2       30865/WHATEVER      # cluster synchronization tool
dircproxy    57000/WHATEVER      # Detachable IRC Proxy
tfido        60177/WHATEVER      # fidonet EMSI over telnet
```

4. As a regular user, search the /etc directory for every file named passwd. Redirect error messages from your search to /dev/null.

Answer:

```
~$ find /etc -name passwd 2>/dev/null
```

```
student@student:~$ find /etc -name passwd 2>/dev/null
/etc/passwd
/etc/cron.daily/passwd
/etc/pam.d/passwd
```

5. Create a directory in your home directory called TEST. Create files in that directory

Answer:

```
~$ mkdir ~/TEST/
```

```
~$ touch ~/TEST/ONE ~/TEST/TWO ~/TEST/THREE
```

```
~$ sudo chmod 777 ~/TEST/ONE ~/TEST/TWO ~/TEST/THREE
```

```
student@student:~$ mkdir ~/TEST/
student@student:~$ touch ~/TEST/ONE ~/TEST/TWO ~/TEST/THREE
student@student:~$ sudo chmod 777 ~/TEST/ONE ~/TEST/TWO ~/TEST/THREE
[sudo] password for student:
```

```
student@student:~$ ls -p ~/TEST/
ONE THREE TWO
```

6. Find files under the /usr/share/doc directory that have not been modified in more than 300 days.

Answer:

```
~$ find /usr/share/doc -mtime +300
```

```
student@student:~$ find /usr/share/doc -mtime +300
```

CT104H – Operating System

```
/usr/share/doc/libunity-scopes-json-def-desktop
/usr/share/doc/libunity-scopes-json-def-desktop/copyright
/usr/share/doc/libunity-scopes-json-def-desktop/changelog.Debian.gz
/usr/share/doc/libxpm4
/usr/share/doc/libxpm4/copyright
/usr/share/doc/libxpm4/changelog.Debian.gz
/usr/share/doc/librtmp1
/usr/share/doc/librtmp1/copyright
/usr/share/doc/librtmp1/changelog.Debian.gz
/usr/share/doc/libsysmetrics1
/usr/share/doc/libsysmetrics1/copyright
/usr/share/doc/libsysmetrics1/changelog.gz
/usr/share/doc/libglvnd0
/usr/share/doc/libglvnd0/copyright
/usr/share/doc/libglvnd0/changelog.Debian.gz
/usr/share/doc/libao-common
/usr/share/doc/libao-common/TODD
/usr/share/doc/libao-common/README
/usr/share/doc/libao-common/copyright
/usr/share/doc/libao-common/changelog.Debian.gz
/usr/share/doc/openprinting-ppds
/usr/share/doc/openprinting-ppds/copyright
/usr/share/doc/openprinting-ppds/changelog.Debian.gz
/usr/share/doc/libmpdec2
/usr/share/doc/libmpdec2/copyright
/usr/share/doc/libmpdec2/changelog.Debian.gz
/usr/share/doc/libmessaging-menu0
/usr/share/doc/libmessaging-menu0/copyright
/usr/share/doc/libmessaging-menu0/changelog.Debian.gz
/usr/share/doc/ubuntu-wallpapers
/usr/share/doc/ubuntu-wallpapers/copyright
/usr/share/doc/ubuntu-wallpapers/changelog.Debian.gz
/usr/share/doc/libxcb-util1
/usr/share/doc/libxcb-util1/README
/usr/share/doc/libxcb-util1/NEWS.gz
/usr/share/doc/libxcb-util1/copyright
/usr/share/doc/libxcb-util1/changelog.Debian.gz
```

7. Create a /tmp/FILES directory. Find all files under the /usr/share directory

that are more than 5MB and less than 10MB and copy them to the /tmp/FILES directory.

Answer:

```
~$ mkdir /tmp/FILES
```

```
~$ find /usr/share/ -size +5mb -and -size -10mb -exec cp {} /tmp/FILES \;
```

```
student@student:~$ mkdir /tmp/FILES
student@student:~$ find /usr/share/ -size +5mb -and -size -10mb -exec cp {} /tmp/FILES/ \;
find: Invalid argument '+5mb' to -size
student@student:~$ find /usr/share/ -size +5M -and -size -10M -exec cp {} /tmp/FILES/ \;
```

```
student@student:~$ ls -p /tmp/FILES/
GeoIPv6.dat  images_tango.zip  NotoColorEmoji.ttf
```

8. Find every file in the /tmp/FILES directory, and make a backup copy of each file in the same directory. Use each file's existing name, and just append .mybackup to create each backup file.

Answer:

```
~$ find /tmp/FILES -type f -exec cp {} {}.mybackup \;
```

```
student@student:~$ find /tmp/FILES/ -type f -exec cp {} {}.mybackup \;
student@student:~$ ls -p /tmp/FILES/
GeoIPv6.dat  GeoIPv6.dat.mybackup  images_tango.zip  images_tango.zip.mybackup  NotoColorEmoji.ttf  NotoColorEmoji.ttf.mybackup
student@student:~$
```

9. Install the kernel-doc package in Fedora or Red Hat Enterprise Linux. Using `grep`, search inside the files contained in the `/usr/share/doc/kernel-doc*` directory for the term `e1000` (case-insensitive) and list the names of the files that contain that term.

Answer:

```
~$ sudo apt install linux-source
```

```
ubuntub2405065@CoseSystem:/usr/src$ sudo apt install linux-source
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu
  gcc-x86-64-linux-gnu libaom3 libasan8 libatomic1 libc-bin libc-dev-bin libc-devtools libc6 libc6-dev libcc1-0
  libcrypt-dev libde265-0 libgcc-13-dev libgd3 libgomp1 libheif-plugin-aomdec libheif-plugin-aomenc
  libheif-plugin-libde265 libheif1 libhwasan0 libisl23 libitm1 liblsan0 libmpc3 libquadmath0 libtsan2 libubsan1
  libxpm4 linux-libc-dev linux-source-6.8.0 locales make manpages-dev rpcsvc-proto
```

```
~$ cd /usr/src
```

```
~$ sudo tar -xvf linux-source-* .tar.bz2
```

```
ubuntub2405065@CoseSystem:/usr/src$ ls /usr/src
linux-source-6.8.0  linux-source-6.8.0.tar.bz2
ubuntub2405065@CoseSystem:/usr/src$ sudo tar -xvf linux-source-* .tar.bz2
ubuntub2405065@CoseSystem:/usr/src$ ls /usr/src/linux-source-*/Documentation
ABI          block        fb            isdn          nvdimmm       subsystem-apis.rst
Changes      bpf          features      kbuild        nvme           target
CodingStyle  cdrom        filesystems   kernel-hacking  pcmcia         tee
Kconfig      cgroups     firmware-guide  leds          peci           timers
Makefile     conf.py      firmware_class  litmus-tests   power          tools
PCI          core-api    fpga          livepatch     process        trace
RAS          cpu-freq    gpu           locking        rust           translations
RCU          crypto      hid           maintainer     scheduler      usb
SubmittingPatches  dev-tools   hmon          memory-barriers.txt  scsi           userspace-api
accel        devicetree  i2c           mhi            security       virt
accounting   doc-guide   iio           misc-devices    sound          w1
admin-guide  docutils.conf  images        mm             sphinx         watchdog
arch         dontdiff    index.rst     netlabel       sphinx-static  wmi
atomic_bitops.txt  driver-api  infiniband    netlink        spi
atomic_t.txt  fault-injection  input         networking     staging
```

```
~$ grep -Ri "e1000" /usr/src/linux-source-* -l
```

```
ubuntub2405065@CoseSystem:/usr/src$ grep -Ri "e1000" /usr/src/linux-source-* -l
/usr/src/linux-source-6.8.0/tools/bpf/bpftool/vmlinux
/usr/src/linux-source-6.8.0/tools/include/uapi/linux/kvm.h
/usr/src/linux-source-6.8.0/tools/testing/selftests/bpf/uprobe_multi.c
/usr/src/linux-source-6.8.0/tools/testing/selftests/sgx/sigstruct.c
/usr/src/linux-source-6.8.0/tools/testing/selftests/arm64/fp/sme-inst.h
/usr/src/linux-source-6.8.0/tools/testing/selftests/arm64/abi/syscall-abi-asm.S
/usr/src/linux-source-6.8.0/sound/soc/sof/mediatek/mt8186/mt8186.h
/usr/src/linux-source-6.8.0/sound/isa/cs423x/cs4236.c
/usr/src/linux-source-6.8.0/sound/pci/hda/patch_ca0132.c
/usr/src/linux-source-6.8.0/arch/riscv/configs/defconfig
/usr/src/linux-source-6.8.0/arch/powerpc/boot/dts/currituck.dts
/usr/src/linux-source-6.8.0/arch/powerpc/boot/dts/fsl/qoriq-fman3-0-1g-0.dtsi
/usr/src/linux-source-6.8.0/arch/powerpc/boot/dts/fsl/qoriq-fman3-0-1g-1.dtsi
/usr/src/linux-source-6.8.0/arch/powerpc/boot/dts/fsl/qoriq-fman3-1-1g-3.dtsi
```

10. Search for the **e100 0** term again in the same location, but this time list every line that contains the term and highlight the term in color

Answer:

```
~$ grep -Ri --color "e1000" /usr/src/linux-source-* -l
```

```
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:    } else if (phy->smart_speed == e1000_sma
rt_speed_off) {
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                IGP
01E1000_PHY_PORT_CONFIG,
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                data &= ~IGP01E1000_PSCFR_SMART_
SPEED;
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                IGP
01E1000_PHY_PORT_CONFIG,
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:    } else if ((phy->autoneg_advertised == E1000_ALL
_SPEED_DUPLEX) ||
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                (phy->autoneg_advertised == E1000_ALL
_NOT_GIG) ||
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                (phy->autoneg_advertised == E1000_ALL
_10_SPEED)) {
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                data |= IGP02E1000_PM_D3_LPLU;
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                ret_val = phy->ops.write_reg(hw, IGP02E1
000_PHY_POWER_MGMT,
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                ret_val = phy->ops.read_reg(hw, IGP01E10
00_PHY_PORT_CONFIG,
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                data &= ~IGP01E1000_PSCFR_SMART_SPEED;
/usr/src/linux-source-6.8.0/drivers/net/ethernet/intel/igb/e1000_phy.c:                                ret_val = phy->ops.write_reg(hw, IGP01E1
000_PHY_PORT_CONFIG,
```

Solve all the questions from 1 to 5 in the Section Exercises of Chapter 7 (Linux Bible of Christopher Negus)

1. Create a script in your \$HOME/bin directory called myownscript. When the script

runs, it should output information that looks as follows:

Today is Sat Dec 10 15:45:04 EST 2016.

You are in /home/joe and your host is abc.example.com.

Of course, you need to read in your current date/time, current working directory,

and hostname. Also, include comments about what the script does and indicate

that the script should run with the /bin/bash shell.

Answer:

```
~$ mkdir -p ~/home/bin
```

```
~$ ls ~/home
```

```
~$ vim ~/home/bin/myownscript
```

```
ubuntub2405065@CoseSystem:~$ ls ~/home  
bin  
ubuntub2405065@CoseSystem:~$ vim ~/home/bin/myownscript
```

```
#!/bin/bash
```

```
# myownscript
```

```
# This script displays the current date and time,
```

```
# the current working directory, and the system hostname.
```

```
CURRENT_DATE=$(date +"Today is %A %m %d at %H:%M:%S %Z %y.")
```

```
CURRENT_DIR=$(pwd)
```

```
HOSTNAME=$(hostname -f)
```

```
echo "$CURRENT_DATE"
```

```
echo "You are in $CURRENT_DIR and your host is $HOSTNAME."
```

```
#!/bin/bash  
# myownscript  
# This script displays the current date and time,  
# the current working directory, and the system hostname.  
CURRENT_DATE=$(date +"Today is %A %m %d at %H:%M:%S %Z %y.")  
CURRENT_DIR=$(pwd)  
HOSTNAME=$(hostname -f)  
echo "$CURRENT_DATE"  
echo "You are in $CURRENT_DIR and your host is $HOSTNAME."
```

```
~$ chmod +x ~/home/bin/myownscript
```

```
~$ ~/home/bin/myownscript
```

```
ubuntub2405065@CoseSystem:~$ chmod +x ~/home/bin/myownscript
```

```
ubuntub2405065@CoseSystem:~$ ~/home/bin/myownscript  
Today is Thursday 01 29 at 10:43:34 UTC 26.  
You are in /home/ubuntub2405065 and your host is CoseSystem.localdomain.  
ubuntub2405065@CoseSystem:~$ |
```

2. Create a script that reads in three positional parameters from the command line,
assigns those parameters to variables named ONE, TWO, and THREE, respectively,

and outputs that information in the following format:

There are X parameters that include Y.

The first is A, the second is B, the third is C.

Answer:

```
~$ vim ~/home/bin/paramscript
```

```
ubuntu2405065@CoseSystem:~$ vim ~/home/bin/paramscript
ubuntu2405065@CoseSystem:~$ |
```

```
#!/bin/bash
```

```
#This script reads three positional parameters
```

```
#The first is A, the second is B, the third is C.
```

```
# /$@/ is all paraments -- /$#/ is number parament in this case is 3
```

```
ONE=$1
```

```
TWO=$2
```

```
THREE=$3
```

```
echo "There are $# parameters that include $@"
```

```
echo "The first is $ONE, the second is $TWO, the third is $THREE."
```

```
#!/bin/bash
#This script reads three positional parameters
#The first is A, the second is B, the third is C.
# /$@/ is all paraments -- /$#/ is number parament in this case is 3
ONE=$1
TWO=$2
THREE=$3
echo "There are $# parameters that include $@"
echo "The first is $ONE, the second is $TWO, the third is $THREE."
~
~
```

```
~$ chmod +x ~/home/bin/paramscript
```

```
~$ ~/home/bin/paramscript apple banana durin
```

```
ubuntub2405065@CoseSystem:~$ ~/home/bin/paramscript apple banana durin
There are 3 parameters that include apple banana durin
The first is apple, the second is banana, the third is durin.
ubuntub2405065@CoseSystem:~$ |
```

3. Create a script that prompts users for the name of the street and town where they grew up. Assign town and street to variables called mytown and mystreet, and output them with a sentence that reads as shown in the following code (of course, \$mystreet and \$mytown will appear with the actual town and street the user enters):

The street I grew up on was \$mystreet and the town was \$mytown.

Answer:

```
~$ vim ~/home/bin/inputsript
```

```
#!/bin/bash
```

```
#This script asks the user for the street and town
```

```
#output them with a sentence that reads as shown in the following code
```

```
echo "input mystreet: "
```

```
read mystreet
```

```
echo "input mytown: "
```

```
read mytown
```

```
echo "The street I grew up on was $mystreet and the town was $mytown"
```

```
#!/bin/bash
#This script asks the user for the street and town
#output them with a sentence that reads as shown in the following code
echo "input mystreet: "
read| mystreet
echo "input mytown: "
read mytown
echo "The street I grew up on was $mystreet and the town was $mytown"
~
```

```
~$ chmod +x ~/home/bin/inputsript
```

```
ubuntu2405065@CoseSystem:~$ ~/home/bin/inputscrip
input mystreet:
Đề Thám
input mytown:
Cần Thơ
The street I grew up on was Đề Thám and the town was Cần Thơ
ubuntu2405065@CoseSystem:~$ |
```

4. Create a script called myos that asks the user, “What is your favorite operating system?” Output an insulting sentence if the user types Windows or Mac. Respond “Great choice!” if the user types Linux. For anything else, say “Is <what is typed in> an operating system?”

Answer:

```
~$ vim ~/home/bin/myos
```

```
#!/bin/bash
```

```
# This script asks the user for their favorite operating system
```

```
# and responds accordingly.
```

```
read -p "What is your favorite operating system?" myos
```

```
if [ "$myos" = "Windows" ] || [ "$myos" = "Mac" ]; then
```

```
    echo "Interesting choice..."
```

```
elif [ "$myos" = "Linux" ]; then
```

```
    echo "Great choice!"
```

```
else
```

```
    echo "Is $myos an operating system?"
```

```
fi
```

```
#!/bin/bash
# This script asks the user for their favorite operating system
# and responds accordingly.
read -p "What is your favorite operating system?" myos
if [ "$myos" = "Windows" ] || [ "$myos" = "Mac" ]; then
    echo "Interesting choice..."
elif [ "$myos" = "Linux" ]; then
    echo "Great choice!"
else
    echo "Is $myos an operating system?"
fi
```

```
~$ chmod +x ~/home/bin/myos
```

```
~$ ~/home/bin/myos
```

case: Windows or Mac

```
ubuntub2405065@CoseSystem:~$ ~/home/bin/myos
What is your favorite operating system? Mac
Interesting choice...
```

case: Linux

```
ubuntub2405065@CoseSystem:~$ ~/home/bin/myos
What is your favorite operating system? Linux
"Great choice!"
```

case: others

```
ubuntub2405065@CoseSystem:~$ ~/home/bin/myos
What is your favorite operating system? blah
"Is blah an operating system?"
ubuntub2405065@CoseSystem:~$ |
```

5. Create a script that runs the words moose, cow, goose, and sow through a for loop.

Have each of those words appended to the end of the line "I have a...."

Answer:

```
~$ vim ~/home/bin/animalscript
```

```
#!/bin/bash
```

```
for animal in moose cow goose sow
```

```
do
```

```
    echo "I have a $animal"
```

```
done
```

```
#!/bin/bash
for animal in moose cow goose sow
do
    echo "I have a $animal"
done
|
~
~
```

~\$ chmod +x ~/home/bin/animalscript

~\$ ~/home/bin/animalscript

```
ubuntub2405065@CoseSystem:~$ chmod +x ~/home/bin/animalscript
ubuntub2405065@CoseSystem:~$ ~/home/bin/animalscript
"I have a moose"
"I have a cow"
"I have a goose"
"I have a sow"
```