

**PROJECT - NGUYÊN LÝ HỆ ĐIỀU HÀNH****Học kỳ 1 – 2023-2024****Máy ảo sử dụng: Ubuntu VM**

Cách thức nộp bài: 01 folder tên mã số sinh viên (**MSSV**) và họ tên của anh/chị **MSSV\_HọTên** chứa

- 01 báo cáo mô tả chi tiết cách giải quyết các vấn đề nêu lên trong project (**kể cả hình ảnh minh họa kết quả và các lệnh** anh/chị thực hiện theo đúng trình tự)
- Tất cả code (cả mức kernel và user) có liên quan

Lưu ý:

- File báo cáo cần có Họ tên và MSSV và phần tuyên bố như bên dưới
- Code cần có chú thích rõ ràng, đầy đủ

**Tuyên bố: Project này là do chính tôi, *Quách Minh Hôn* (MSSV: **B2110078**), tự thực hiện, không sao chép của bất kỳ ai. Nếu có bất cứ sao chép nào, tôi hoàn toàn chịu trách nhiệm.**

---

## **PHẦN 1: XÂY DỰNG LINUX KERNEL**

Thực hiện ở chế độ người dùng root (root user)

**\$su -**

### **A. CHUẨN BỊ LINUX KERNEL CODE (phần hướng dẫn dưới đây thực hiện trên CentOS. Tuy nhiên, sinh viên cần thực hiện trên Ubuntu)**

#### **1. Download và cài đặt tool cần thiết vào hệ thống**

Nếu thực hiện ở CentOS:

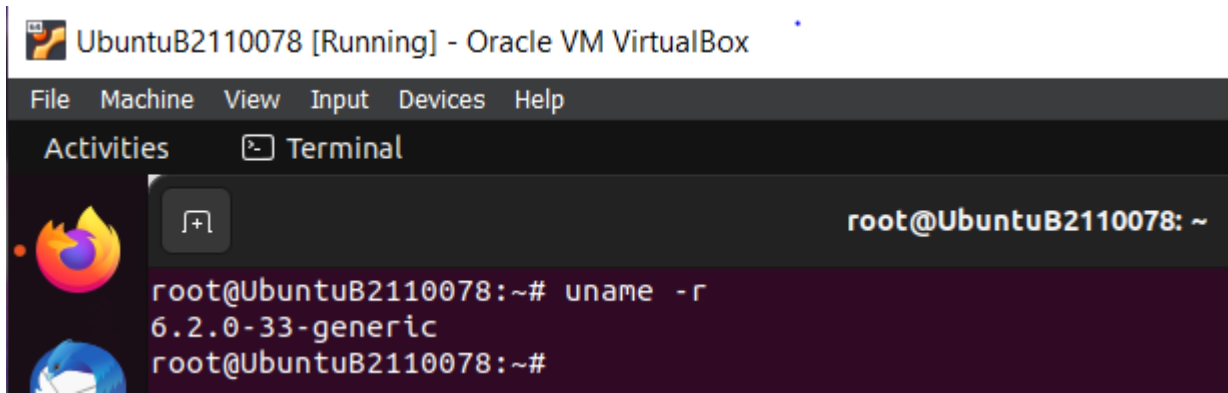
```
#yum install -y gcc ncurses-devel make wget  
#yum apt-get install -y gcc openssl-devel  
#yum apt-get install -y gcc elfutils-libelf-devel  
#yum apt-get install bison  
#yum apt-get install flex
```

Nếu thực hiện ở Ubuntu:

```
$sudo apt-get install -y gcc libncurses5-dev make wget  
$sudo apt-get install -y gcc libssl-dev  
$sudo apt-get install bison  
$sudo apt-get install flex
```

#### **2. Xác định phiên bản hiện tại của kernel:**

```
# uname -r  
6.2.0-33-generic
```

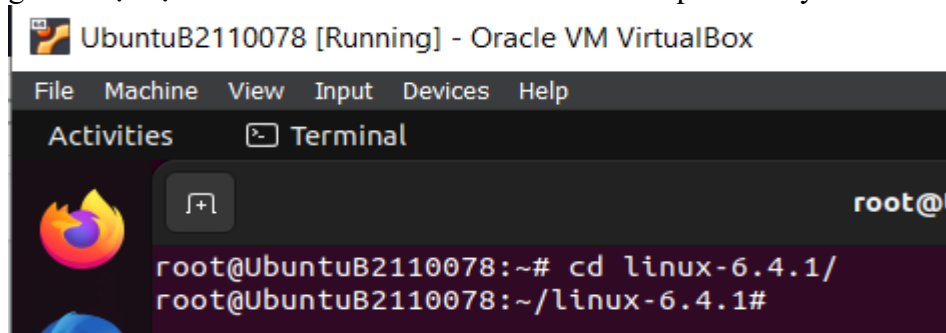


3. Truy cập <http://kernel.org> hoặc <https://cdn.kernel.org/pub/linux/kernel/> và download source code của kernel hiện tại. Kế tiếp, download kernel 5.9 và giải nén source code:

```
# wget http://www.kernel.org/pub/linux/kernel/v6.x/linux-6.4.1.tar.xz
# tar xf linux-6.4.1.tar.xz
```

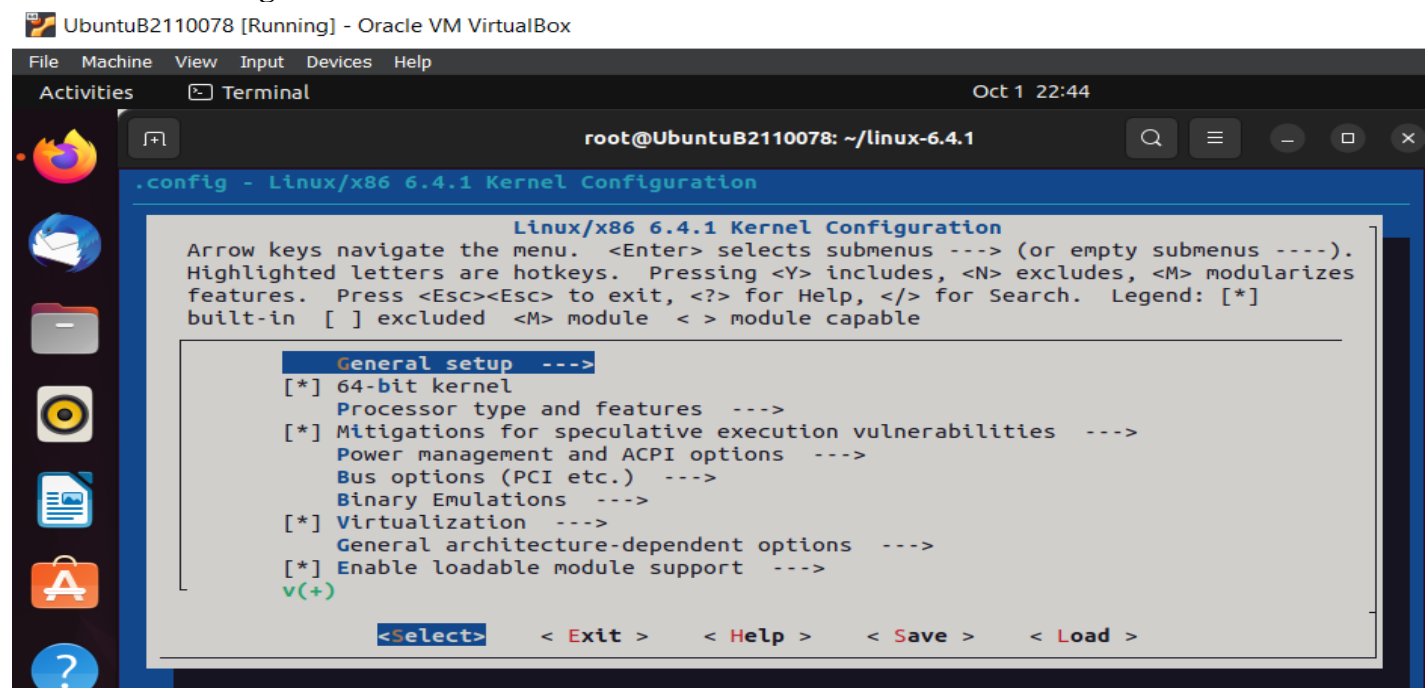
## B. CẤU HÌNH KERNEL MỚI

1. Đảm bảo đường dẫn hiện tại ở `~/linux-6.4.1` và “`linux-6.4.1`” ở top directory của kernel source.



2. Tạo file cấu hình (config file)

```
# make menuconfig
```

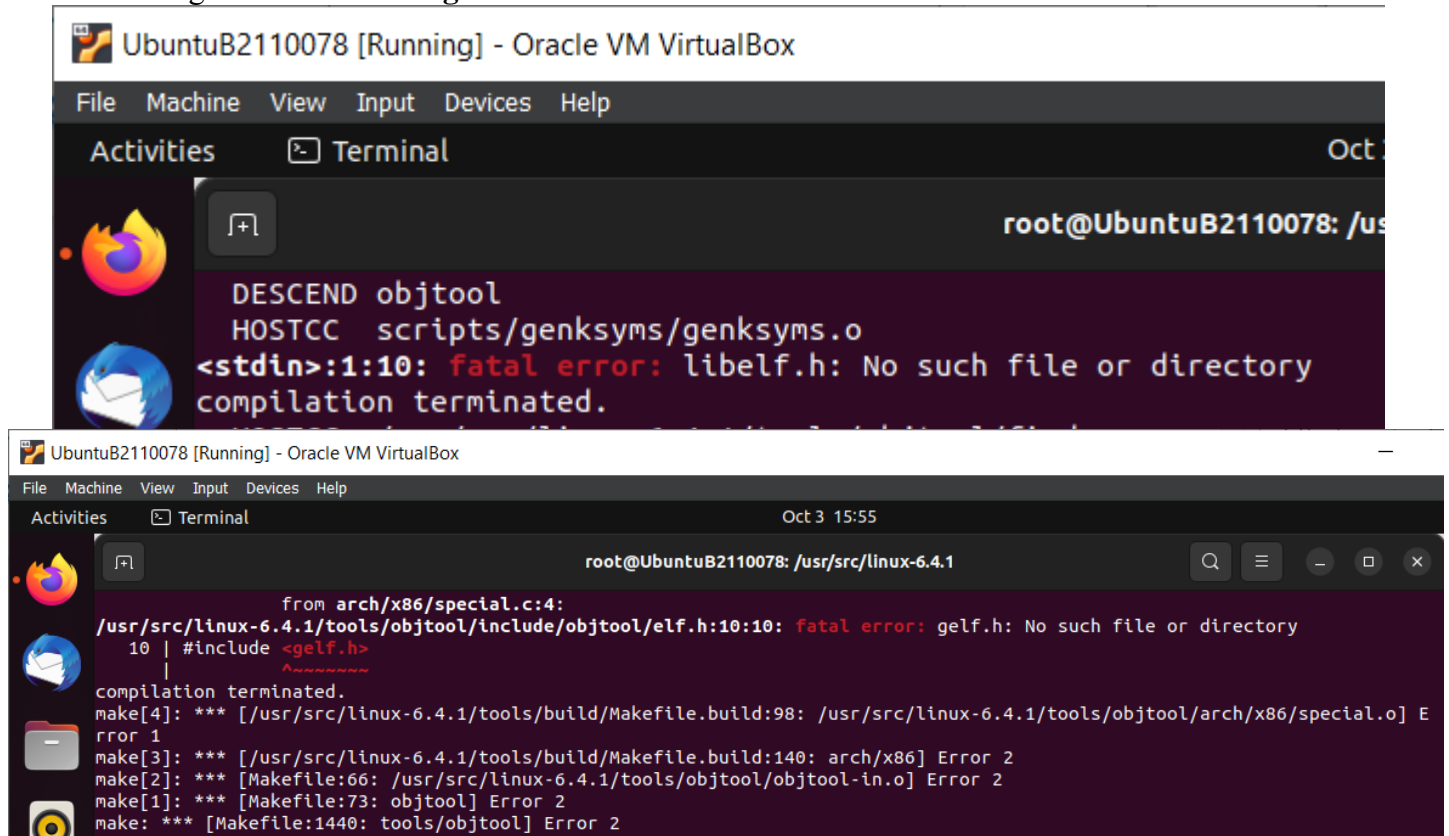


### C. BIÊN DỊCH KERNEL

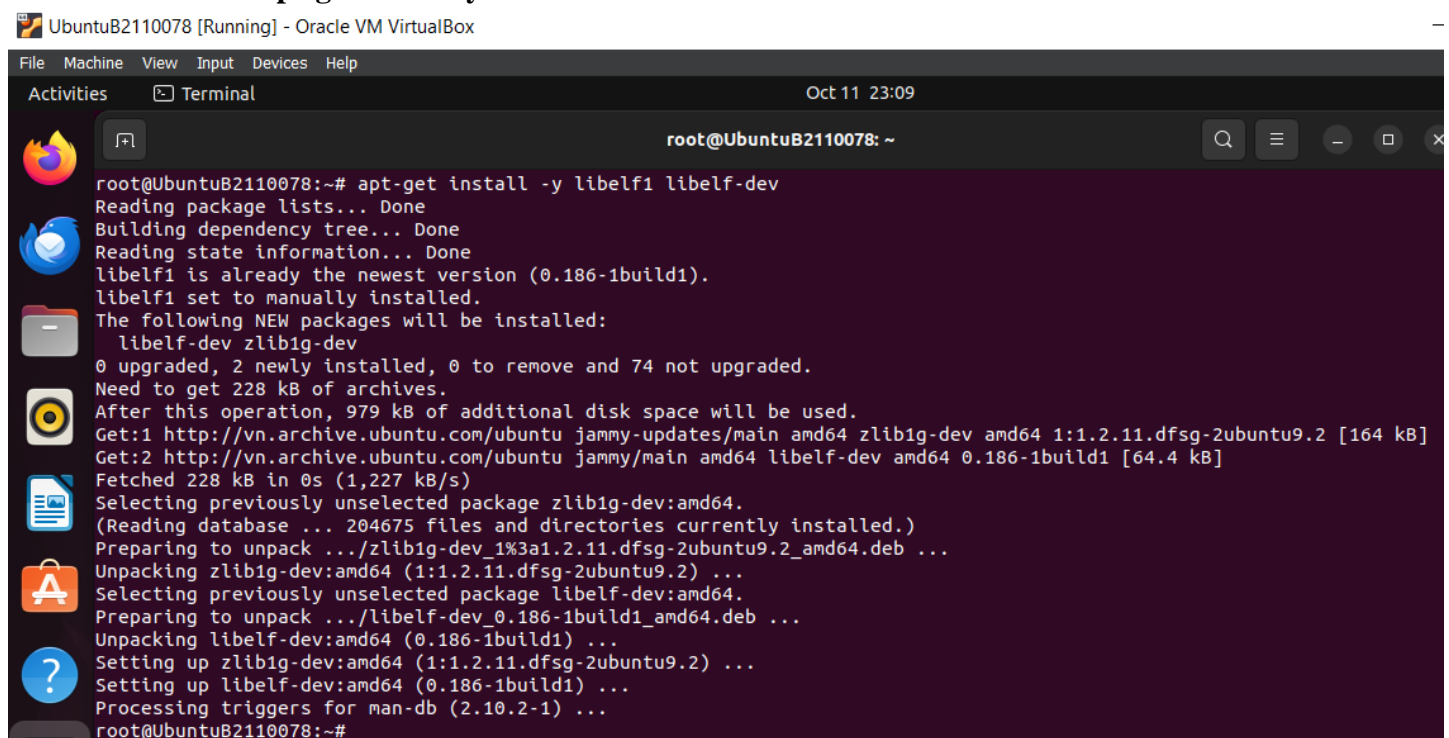
1. Tại ~/linux-5.9, tạo kernel image nén (compressed kernel image)

# make -j4

- Lỗi: Không có file **libelf.h** và **gelf.h**



- Sửa lỗi: # **sudo apt-get install -y libelf1 libelf-dev**



- Lỗi: No rule to make target '**debian/cannonical-certs.pem**'

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```

File  Machine  View  Input  Devices  Help
Activities  Terminal
root@UbuntuB2110078: ~/linux-6.4.1

CC      kernel/events/core.o
make[2]: *** No rule to make target 'debian/cannonical-certs.pem', needed by 'certs/x509_certificate_list'. Stop.
make[2]: *** Waiting for unfinished jobs....
CC      certs/system_keyring.o
AR      kernel/bpf/built-in.a
CC      kernel/fork.o
make[1]: *** [scripts/Makefile.build:494: certs] Error 2
make[1]: *** Waiting for unfinished jobs....

```

- Sửa lỗi: Xóa giá trị của CONFIG\_SYSTEM\_TRUSTED\_KEYS và CONFIG\_SYSTEM\_REVOCATION\_KEYS

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```

File  Machine  View  Input  Devices  Help
Activities  Terminal
root@UbuntuB2110078: /usr

GNU nano 6.2 .confi
CONFIG_MODULE_SIG_KEY_TYPE_RSA=y
# CONFIG_MODULE_SIG_KEY_TYPE_ECDSA is not set
CONFIG_SYSTEM_TRUSTED_KEYRING=y
CONFIG_SYSTEM_TRUSTED_KEYS="debian/cannonical-certs.pem"
CONFIG_SYSTEM_EXTRA_CERTIFICATE=y
CONFIG_SYSTEM_EXTRA_CERTIFICATE_SIZE=4096
CONFIG_SECONDARY_TRUSTED_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_HASH_LIST=""
CONFIG_SYSTEM_REVOCATION_LIST=y
CONFIG_SYSTEM_REVOCATION_KEYS="debian/cannonical-revoked-certs.pem"

```

UbuntuB2110078 [Running] - Oracle VM VirtualBox

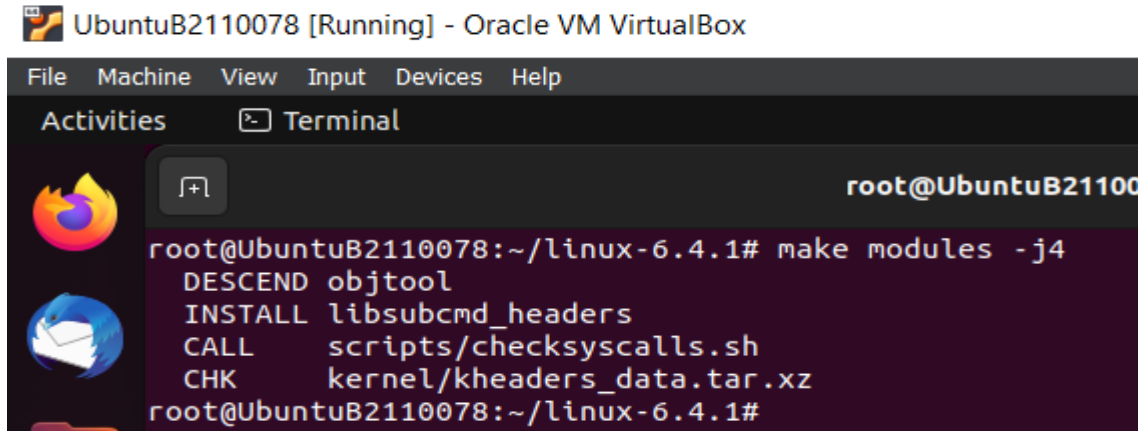
```

File  Machine  View  Input  Devices  Help
Activities  Terminal
root@Ubuntu

GNU nano 6.2
#
CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
CONFIG_MODULE_SIG_KEY_TYPE_RSA=y
# CONFIG_MODULE_SIG_KEY_TYPE_ECDSA is not set
CONFIG_SYSTEM_TRUSTED_KEYRING=y
CONFIG_SYSTEM_TRUSTED_KEYS=""
CONFIG_SYSTEM_EXTRA_CERTIFICATE=y
CONFIG_SYSTEM_EXTRA_CERTIFICATE_SIZE=4096
CONFIG_SECONDARY_TRUSTED_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_HASH_LIST=""
CONFIG_SYSTEM_REVOCATION_LIST=y
CONFIG_SYSTEM_REVOCATION_KEYS=""

```

## 2. Biên dịch kernel modules:

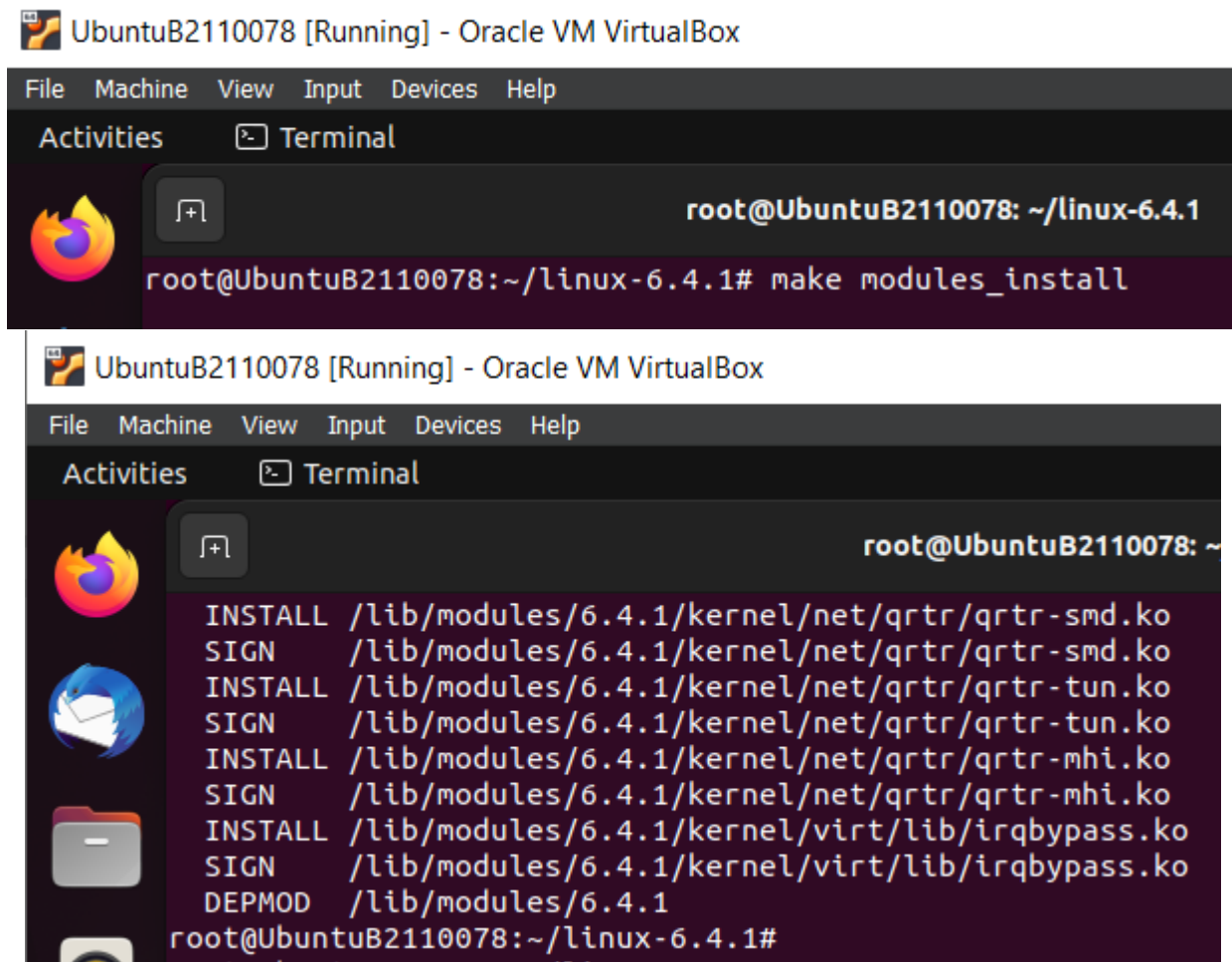
# **make modules**

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities Terminal
root@UbuntuB2110078:~/linux-6.4.1# make modules -j4
DESCEND objtool
INSTALL libsubcmd_headers
CALL scripts/checksyscalls.sh
CHK kernel/kheaders_data.tar.xz
root@UbuntuB2110078:~/linux-6.4.1#
```

## D. CÀI ĐẶT KERNEL

## 1. Cài đặt kernel modules

# **make modules\_install**

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities Terminal
root@UbuntuB2110078: ~/linux-6.4.1
root@UbuntuB2110078:~/linux-6.4.1# make modules_install
```

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities Terminal
root@UbuntuB2110078: ~
INSTALL /lib/modules/6.4.1/kernel/net/qrtr/qrtr-smd.ko
SIGN /lib/modules/6.4.1/kernel/net/qrtr/qrtr-smd.ko
INSTALL /lib/modules/6.4.1/kernel/net/qrtr/qrtr-tun.ko
SIGN /lib/modules/6.4.1/kernel/net/qrtr/qrtr-tun.ko
INSTALL /lib/modules/6.4.1/kernel/net/qrtr/qrtr-mhi.ko
SIGN /lib/modules/6.4.1/kernel/net/qrtr/qrtr-mhi.ko
INSTALL /lib/modules/6.4.1/kernel/virt/lib/irqbypass.ko
SIGN /lib/modules/6.4.1/kernel/virt/lib/irqbypass.ko
DEPMOD /lib/modules/6.4.1
root@UbuntuB2110078:~/linux-6.4.1#
```

## 2. Cài đặt the kernel

# **make install**



UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities Terminal Oct 11 21:54
root@UbuntuB2110078: ~/linux-6.4.1
root@UbuntuB2110078:~/linux-6.4.1# make install
INSTALL /boot
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 6.4.1 /boot/vmlinuz-6.4.1
update-initramfs: Generating /boot/initrd.img-6.4.1
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 6.4.1 /boot/vmlinuz-6.4.1
run-parts: executing /etc/kernel/postinst.d/update-notifier 6.4.1 /boot/vmlinuz-6.4.1
run-parts: executing /etc/kernel/postinst.d/xx-update-initrd-links 6.4.1 /boot/vmlinuz-6.4.1
I: /boot/initrd.img.old is now a symlink to initrd.img-6.2.0-33-generic
I: /boot/initrd.img is now a symlink to initrd.img-6.4.1
run-parts: executing /etc/kernel/postinst.d/zz-shim 6.4.1 /boot/vmlinuz-6.4.1
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.4.1 /boot/vmlinuz-6.4.1
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.4.1
Found initrd image: /boot/initrd.img-6.4.1
Found linux image: /boot/vmlinuz-6.2.0-33-generic
Found initrd image: /boot/initrd.img-6.2.0-33-generic
Found linux image: /boot/vmlinuz-6.2.0-26-generic
Found initrd image: /boot/initrd.img-6.2.0-26-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
root@UbuntuB2110078:~/linux-6.4.1#
```

### E. THAY ĐỔI FILE CẤU HÌNH GRUB (GRUB CONFIGURATION FILE)

Thay đổi file cấu hình grub:

# **vim** /etc/default/grub

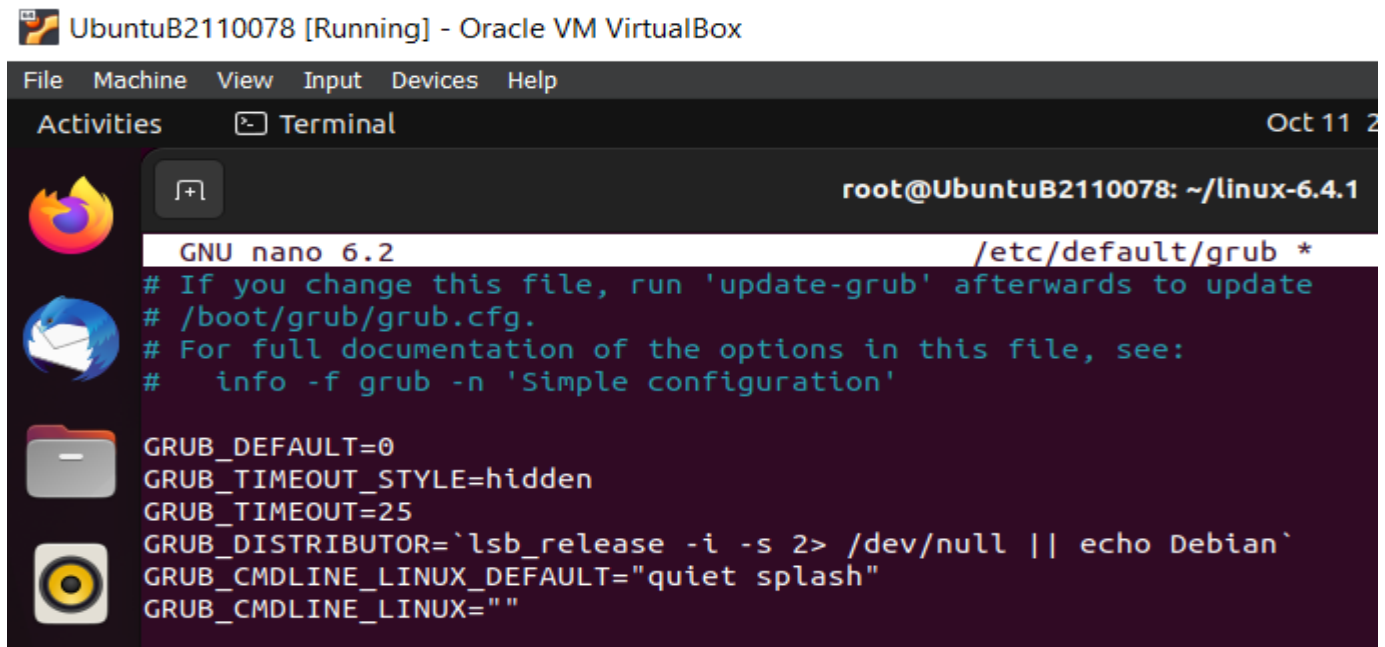
UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities Terminal
root@UbuntuB2110078: ~/lin
root@UbuntuB2110078:~/linux-6.4.1# nano /etc/default/grub
```

Thực hiện các thay đổi sau:

GRUB\_DEFAULT=0

GRUB\_TIMEOUT=25



UbuntuB2110078 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Oct 11 2

root@UbuntuB2110078: ~/linux-6.4.1

```
GNU nano 6.2 /etc/default/grub *
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=25
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

## F. REBOOT VM

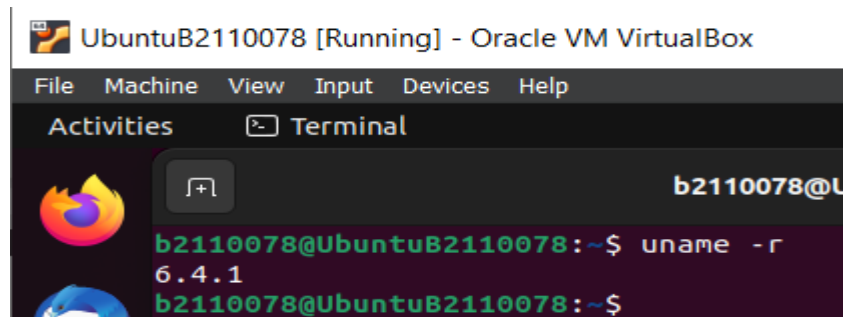
1. Reboot kernel mới:

# **reboot**

2. Sau khi boot, kiểm tra thông tin kernel mới có đúng chưa:

# **uname -r**

Kết quả: 6.4.1



UbuntuB2110078 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal

b2110078@U

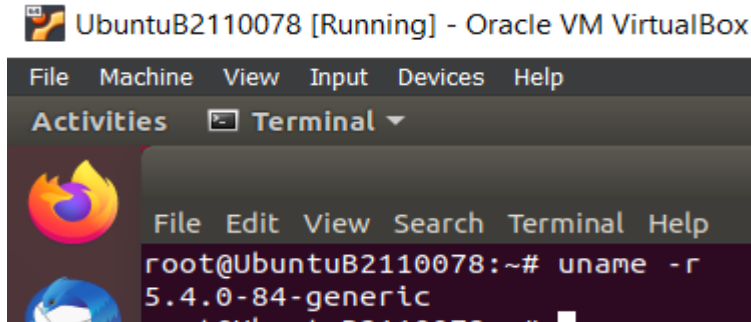
```
b2110078@UbuntuB2110078:~$ uname -r
6.4.1
b2110078@UbuntuB2110078:~$
```

## PHẦN 2: THÊM LỜI GỌI HỆ THỐNG VÀO LINUX KERNEL

Thực hiện thêm lời gọi hệ thống **helloworld** vào Linux kernel. Lời gọi hệ thống thực hiện in ra thông điệp “Xin chào. Ten toi la XXX” vào syslog (XXX là tên và MSSV của anh/chị). Anh/chị cần cài đặt lời gọi hệ thống ở mức kernel và viết một chương trình ở mức người dùng (user-level) để kiểm tra lời gọi hệ thống đã tạo.

### A. CHUẨN BỊ KERNEL SOURCE CODE

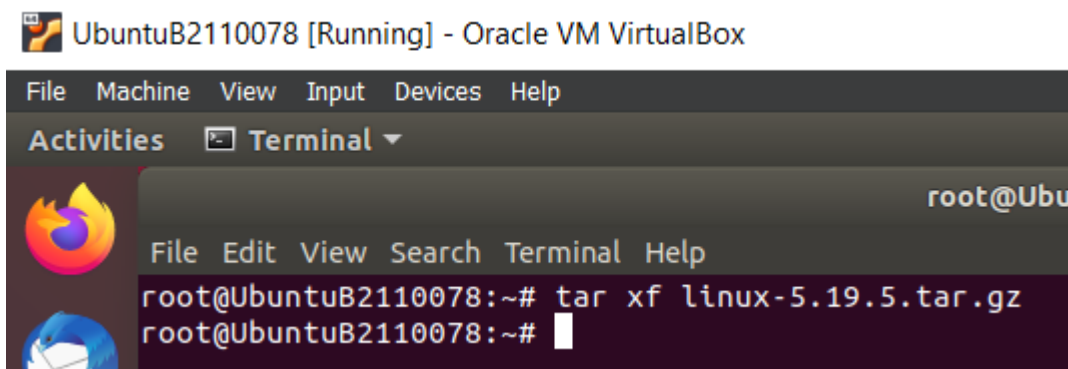
- Xác định phiên bản kernel hiện tại là 5.4.0-84-generic



- Tải xuống kernel phiên bản 5.19.5 và giải nén

# **wget** <https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.5.tar.gz>

# **tar xf linux-5.19.5.tar.gz**






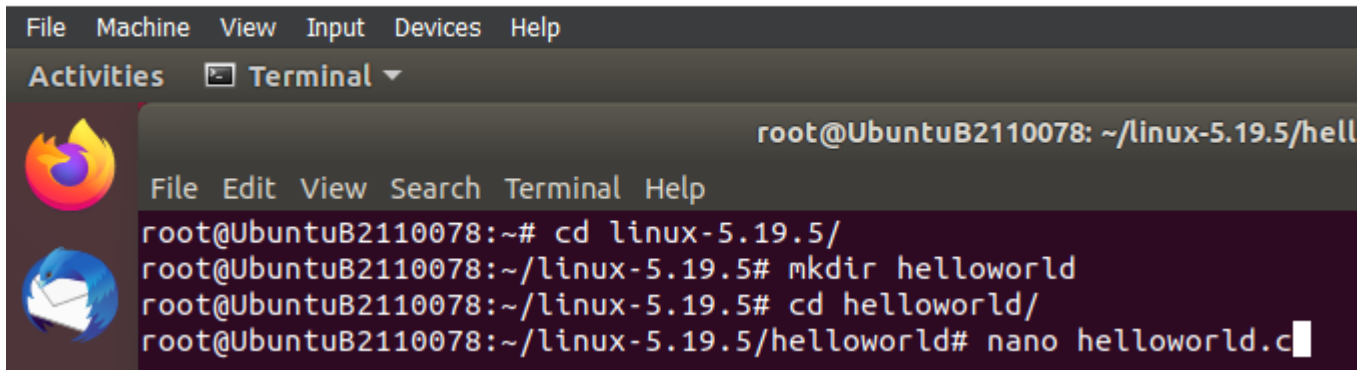
## B. THIẾT LẬP SYSCALL HELLOWORLD

- cd vào thư mục linux-5.19.5, tạo thư mục helloworld
- Trong thư mục helloworld, tạo file helloworld.c với đoạn code bên dưới


```
#include <linux/kernel.h>
#include <linux/syscalls.h>
```

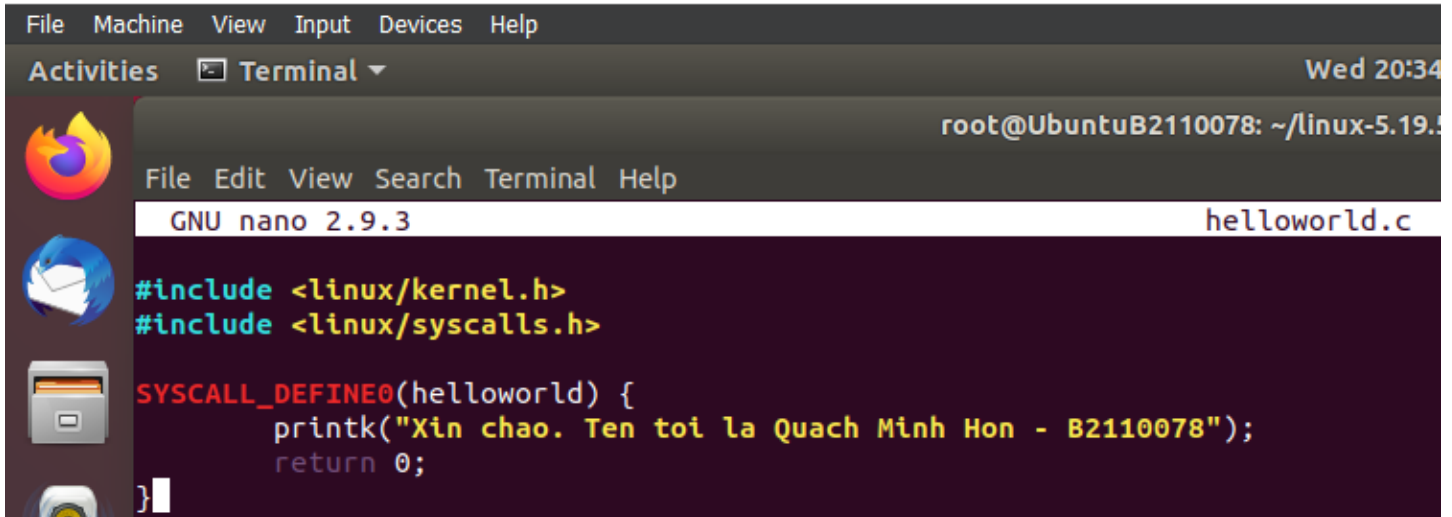
```
SYSCALL_DEFINE0(helloworld) {
    printk("Xin chào. Ten toi la Quach Minh Hon – B2110078");
    return 0;
}
```

 UbuntuB2110078 [Running] - Oracle VM VirtualBox



```
File Machine View Input Devices Help
Activities Terminal
root@UbuntuB2110078: ~/linux-5.19.5/hell
File Edit View Search Terminal Help
root@UbuntuB2110078:~# cd linux-5.19.5/
root@UbuntuB2110078:~/linux-5.19.5# mkdir helloworld
root@UbuntuB2110078:~/linux-5.19.5# cd helloworld/
root@UbuntuB2110078:~/linux-5.19.5/helloworld# nano helloworld.c
```

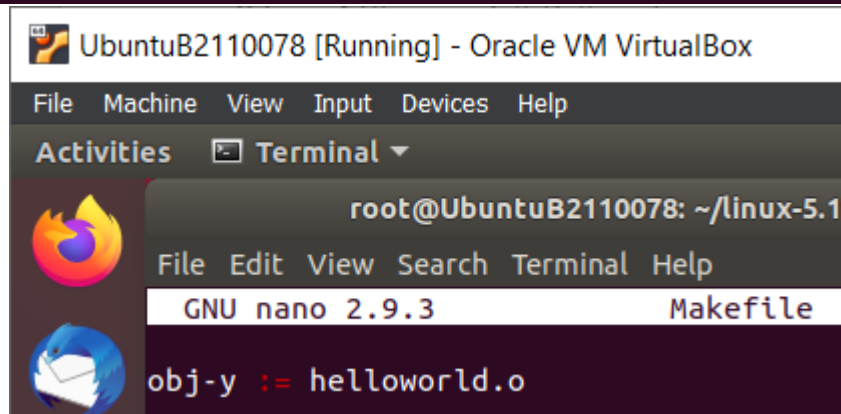
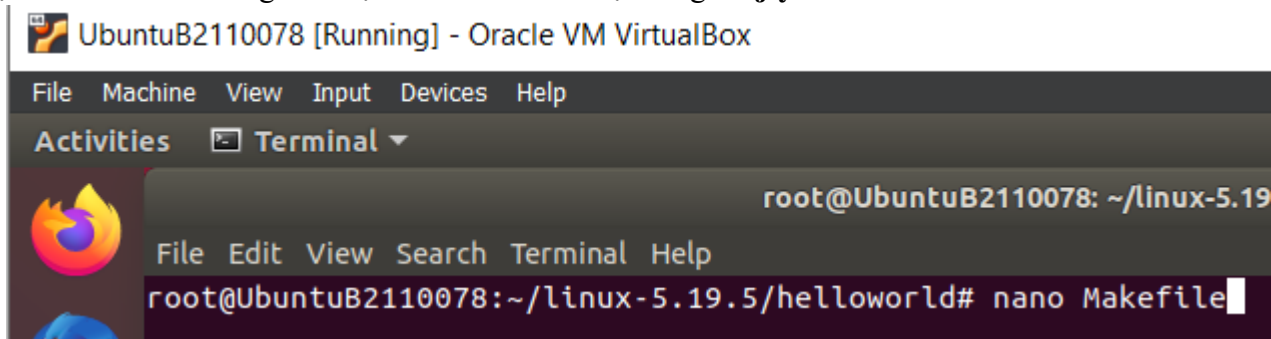
 UbuntuB2110078 [Running] - Oracle VM VirtualBox



```
File Machine View Input Devices Help
Activities Terminal Wed 20:34
root@UbuntuB2110078: ~/linux-5.19.5
File Edit View Search Terminal Help
GNU nano 2.9.3 helloworld.c
#include <linux/kernel.h>
#include <linux/syscalls.h>

SYSCALL_DEFINE0(helloworld) {
    printk("Xin chào. Ten toi la Quach Minh Hon - B2110078");
    return 0;
}
```

- Tạo file Makefile trong thư mục helloworld với nội dung: **obj-y := helloworld.o**

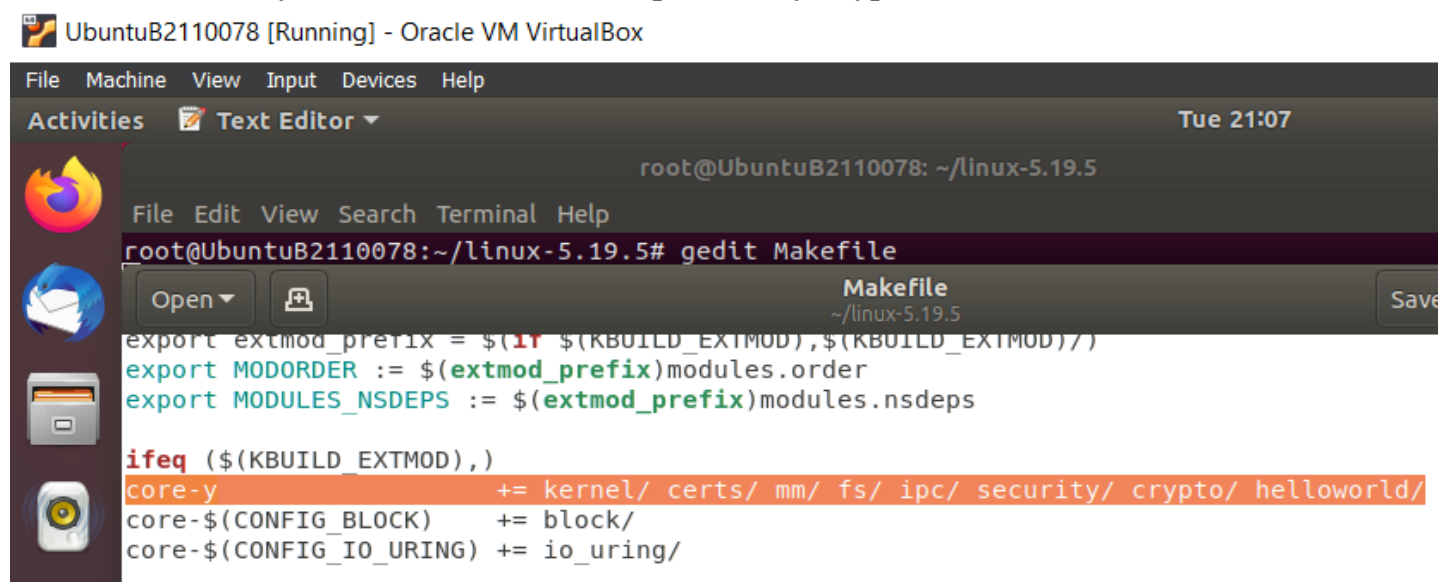


- cd về thư mục linux-5.19.5 để chỉnh sửa file Makefile có sẵn

# cd ..

# gedit Makefile

- Tìm dòng “core-y += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/”
- Sửa thành “core-y += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ helloworld/”



- Mở file `include/linux/syscalls.h` để chèn lời gọi hệ thống mới vào files syscalls header: **asmlinkage long sys\_helloworld(void);**

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```

File Machine View Input Devices Help
Activities Terminal
root@UbuntuB2110078: ~/linux-5.19.5
File Edit View Search Terminal Help
root@UbuntuB2110078:~/linux-5.19.5# nano include/linux/syscalls.h
root@UbuntuB2110078:~/linux-5.19.5#
  
```

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```

File Machine View Input Devices Help
Activities Terminal Tue 21:13
root@UbuntuB2110078: ~/linux-5.19.5
File Edit View Search Terminal Help
GNU nano 2.9.3 include/linux/syscalls.h
long __do_senrtimedop(int semid, struct sembuf *tsems, unsigned int nsops,
                    const struct timespec64 *timeout,
                    struct ipc_namespace *ns);
int __sys_getsockopt(int fd, int level, int optname, char __user *optval,
                    int __user *optlen);
int __sys_setsockopt(int fd, int level, int optname, char __user *optval,
                    int optlen);
asmlinkage long sys_helloworld(void);
#endif
  
```

- Mở file `arch/x86/entry/syscalls/syscall_64.tbl`
- Chèn lời gọi hệ thống mới vào syscalls table:  
**451 common helloworld sys\_helloworld**

UbuntuB2110078 [Running] - Oracle VM VirtualBox

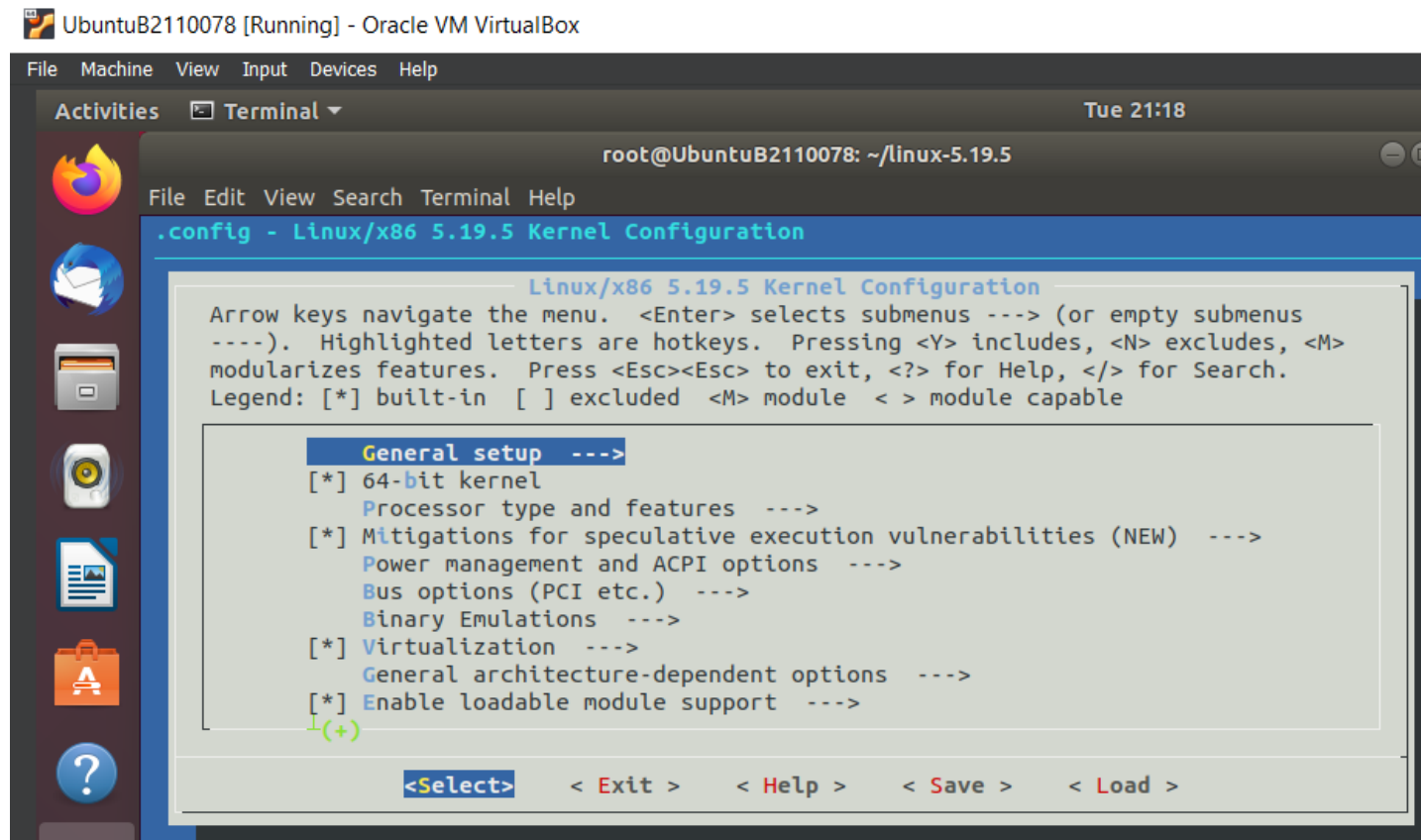
```

File Machine View Input Devices Help
Activities Text Editor Tue 21:17
root@UbuntuB2110078: ~/linux-5.19.5
File Edit View Search Terminal Help
root@UbuntuB2110078:~/linux-5.19.5# gedit arch/x86/entry/syscalls/syscall_64.tbl
syscall_64.tbl
~/linux-5.19.5/arch/x86/entry/syscalls
445 common landlock_add_rule sys_landlock_add_rule
446 common landlock_restrict_self sys_landlock_restrict_self
447 common memfd_secret sys_memfd_secret
448 common process_mrelease sys_process_mrelease
449 common futex_waitv sys_futex_waitv
450 common set_mempolicy_home_node sys_set_mempolicy_home_node
451 common helloworld sys_helloworld
  
```

## C. THIẾT LẬP CẤU HÌNH KERNEL MỚI

- Tạo file cấu hình

# **make menuconfig**

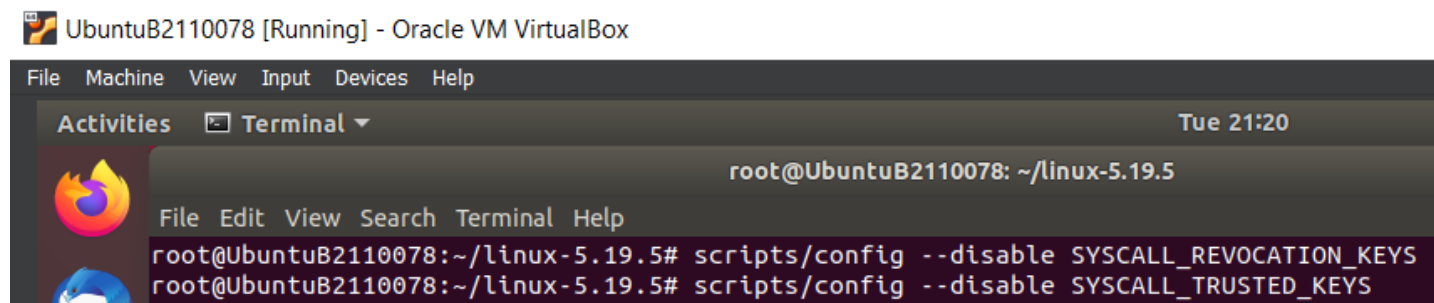


- Chạy lệnh vô hiệu hóa tính năng thu hồi lỗi gọi hệ thống

# **scripts/config --disable --SYSCALL\_REVOCATION\_KEYS**

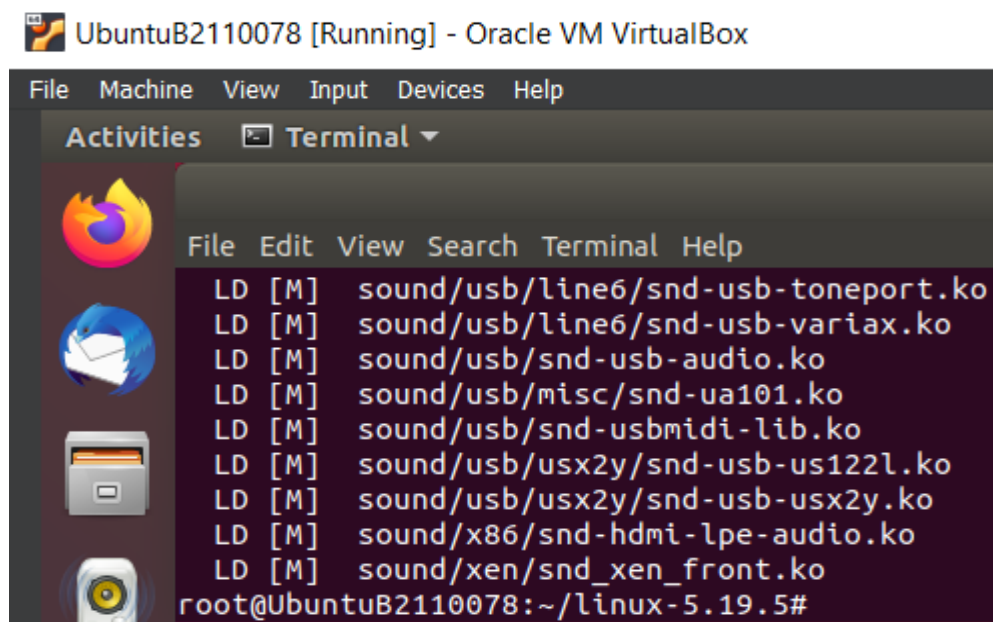
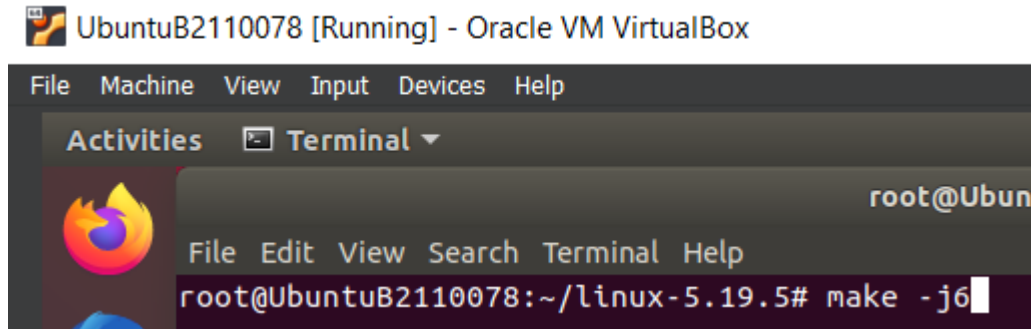
- Chạy lệnh vô hiệu hóa tính năng xác thực chìa khóa

# **scripts/config --disable --SYSCALL\_TRUSTED\_KEYS**



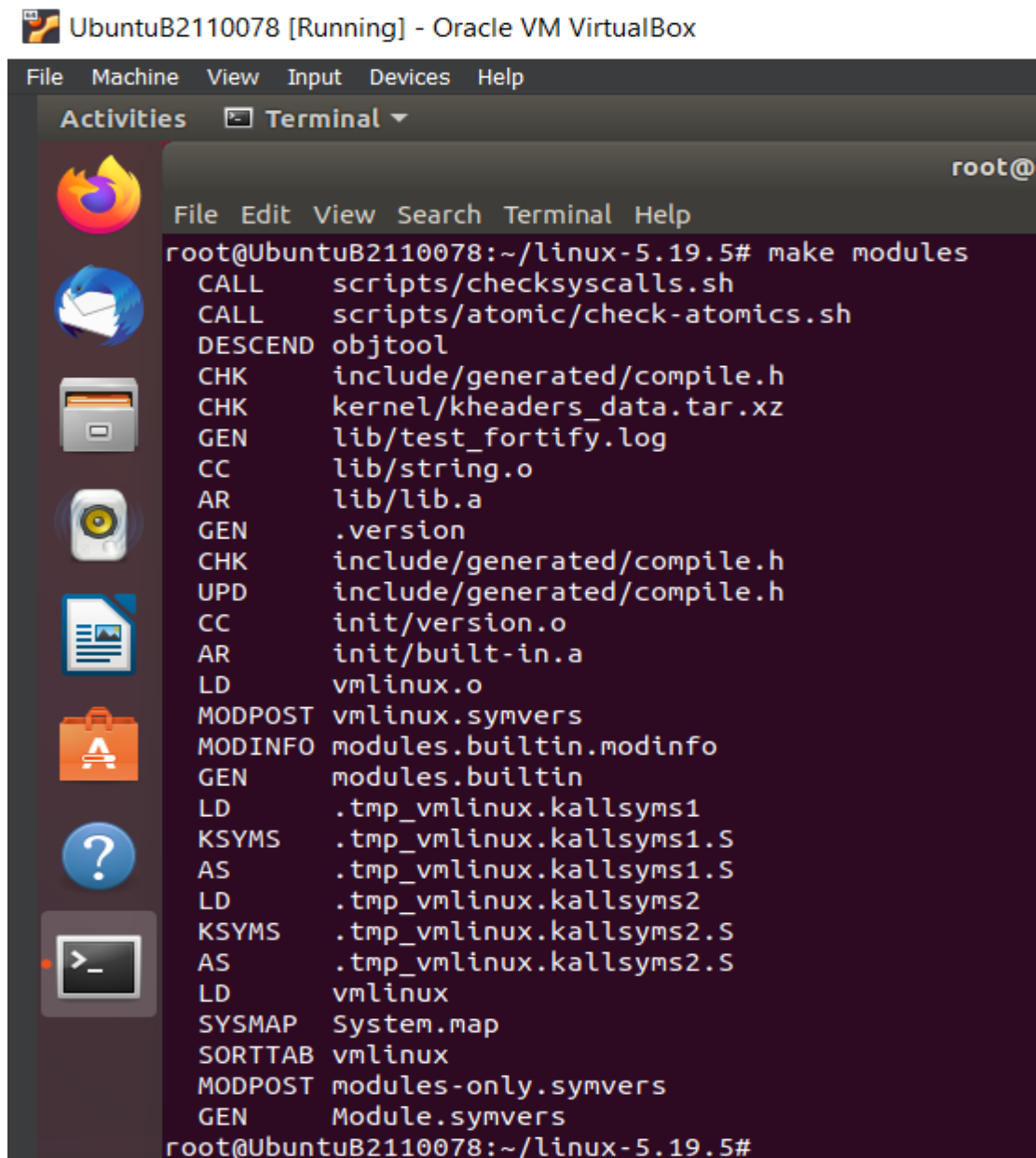
## D. BIÊN DỊCH KERNEL

- Tại ~/linux-5.19.5, tạo kernel image nén  
# **make -j6**



- Biên dịch kernel modules  
# **make modules**





UbuntuB2110078 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

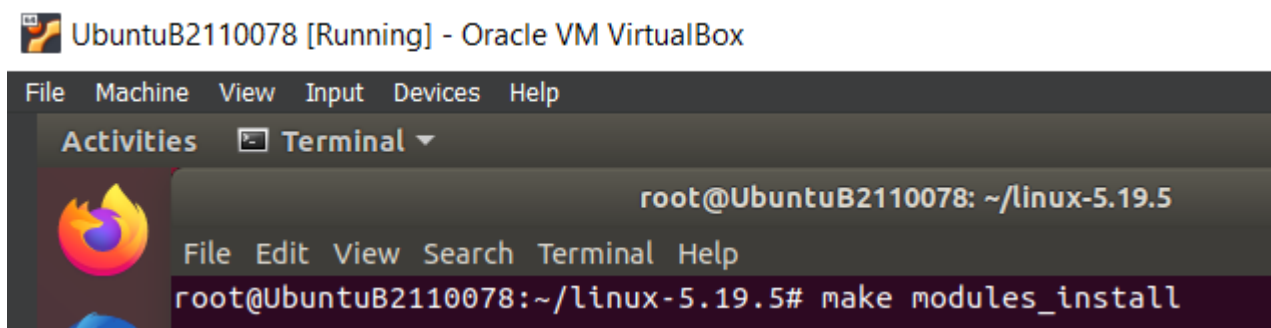
Activities Terminal

root@UbuntuB2110078:~/linux-5.19.5# make modules

```
CALL      scripts/checksyscalls.sh
CALL      scripts/atomic/check-atomics.sh
DESCEND    objtool
CHK        include/generated/compile.h
CHK        kernel/kheaders_data.tar.xz
GEN        lib/test_fortify.log
CC         lib/string.o
AR         lib/lib.a
GEN        .version
CHK        include/generated/compile.h
UPD        include/generated/compile.h
CC         init/version.o
AR         init/built-in.a
LD         vmlinux.o
MODPOST    vmlinux.symvers
MODINFO    modules.builtin.modinfo
GEN        modules.builtin
LD         .tmp_vmlinux.kallsyms1
KSYMS      .tmp_vmlinux.kallsyms1.S
AS         .tmp_vmlinux.kallsyms1.S
LD         .tmp_vmlinux.kallsyms2
KSYMS      .tmp_vmlinux.kallsyms2.S
AS         .tmp_vmlinux.kallsyms2.S
LD         vmlinux
SYSMAP     System.map
SORTTAB    vmlinux
MODPOST    modules-only.symvers
GEN        Module.symvers
root@UbuntuB2110078:~/linux-5.19.5#
```

- Cài đặt kernel modules

# **make modules\_install**



UbuntuB2110078 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal

root@UbuntuB2110078: ~/linux-5.19.5

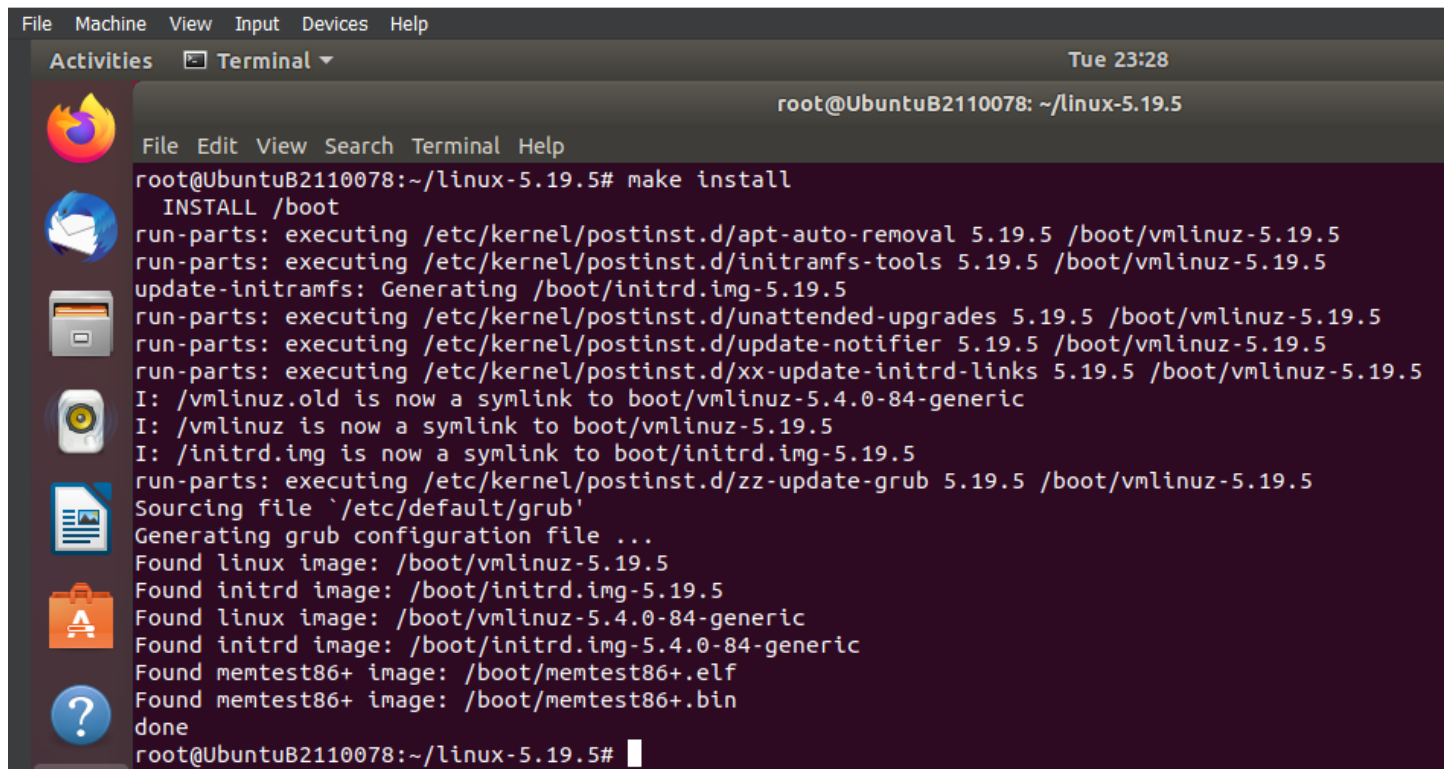
File Edit View Search Terminal Help

root@UbuntuB2110078:~/linux-5.19.5# make modules\_install

- Cài đặt kernel

# **make install**

UbuntuB2110078 [Running] - Oracle VM VirtualBox



```

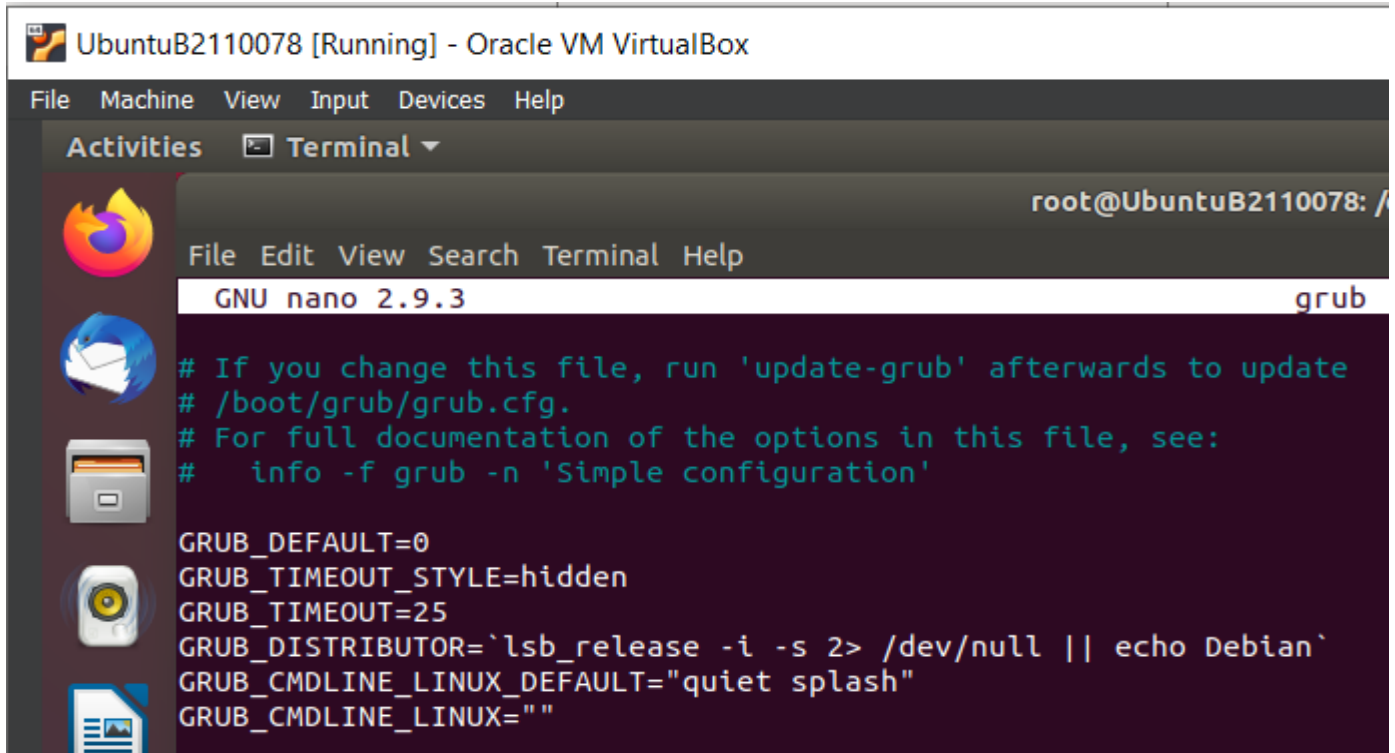
root@UbuntuB2110078: ~/linux-5.19.5
File Edit View Search Terminal Help
root@UbuntuB2110078:~/linux-5.19.5# make install
INSTALL /boot
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.19.5 /boot/vmlinuz-5.19.5
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.19.5 /boot/vmlinuz-5.19.5
update-initramfs: Generating /boot/initrd.img-5.19.5
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.19.5 /boot/vmlinuz-5.19.5
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.19.5 /boot/vmlinuz-5.19.5
run-parts: executing /etc/kernel/postinst.d/xx-update-initrd-links 5.19.5 /boot/vmlinuz-5.19.5
I: /vmlinuz.old is now a symlink to boot/vmlinuz-5.4.0-84-generic
I: /vmlinuz is now a symlink to boot/vmlinuz-5.19.5
I: /initrd.img is now a symlink to boot/initrd.img-5.19.5
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.19.5 /boot/vmlinuz-5.19.5
Sourcing file `/etc/default/grub'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.19.5
Found initrd image: /boot/initrd.img-5.19.5
Found linux image: /boot/vmlinuz-5.4.0-84-generic
Found initrd image: /boot/initrd.img-5.4.0-84-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
root@UbuntuB2110078:~/linux-5.19.5#

```

- Thay đổi cấu hình grub:

GRUB\_DEFAULT=0

GRUB\_TIMEOUT=25



```

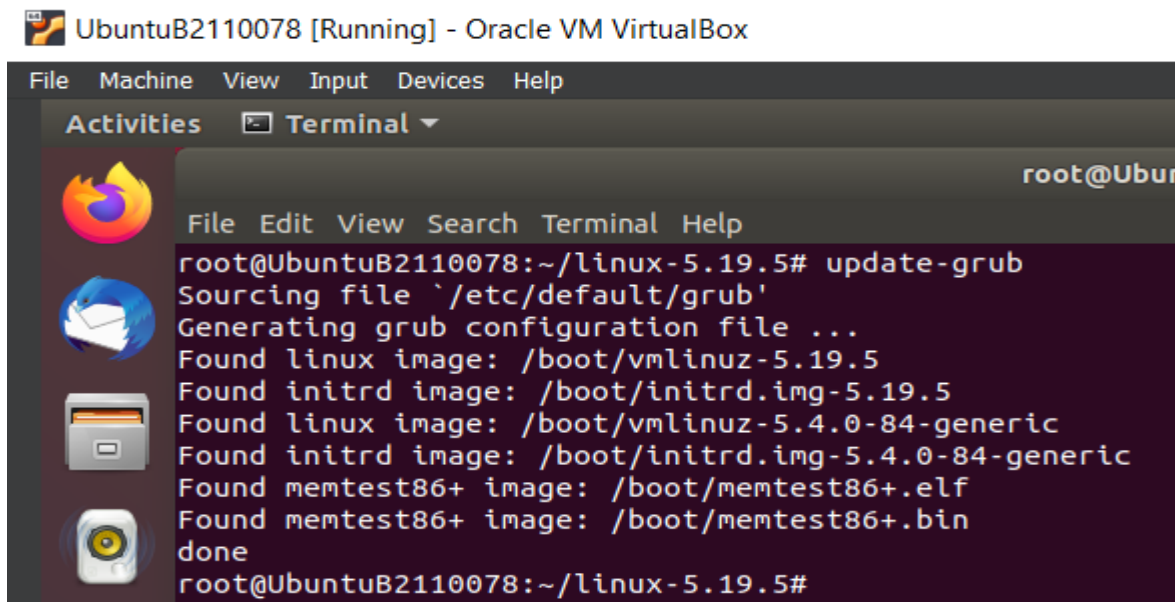
UbuntuB2110078 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
root@UbuntuB2110078: /
File Edit View Search Terminal Help
GNU nano 2.9.3 grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=25
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

```

- Cập nhật cấu hình grub

# **update-grub**



```

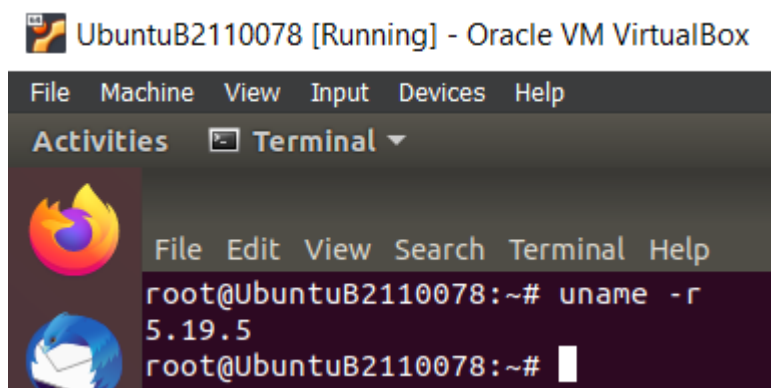
root@UbuntuB2110078:~/linux-5.19.5# update-grub
Sourcing file `/etc/default/grub'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.19.5
Found initrd image: /boot/initrd.img-5.19.5
Found linux image: /boot/vmlinuz-5.4.0-84-generic
Found initrd image: /boot/initrd.img-5.4.0-84-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
root@UbuntuB2110078:~/linux-5.19.5#

```

- Reboot kernel mới và kiểm tra phiên bản kernel

# **reboot**

# **uname -r**



```

root@UbuntuB2110078:~# uname -r
5.19.5
root@UbuntuB2110078:~#

```

## E. THỬ NGHIỆM SYSCALL MỚI

- Tạo file hello.c với đoạn code bên dưới:

```
#include <linux/kernel.h>
```

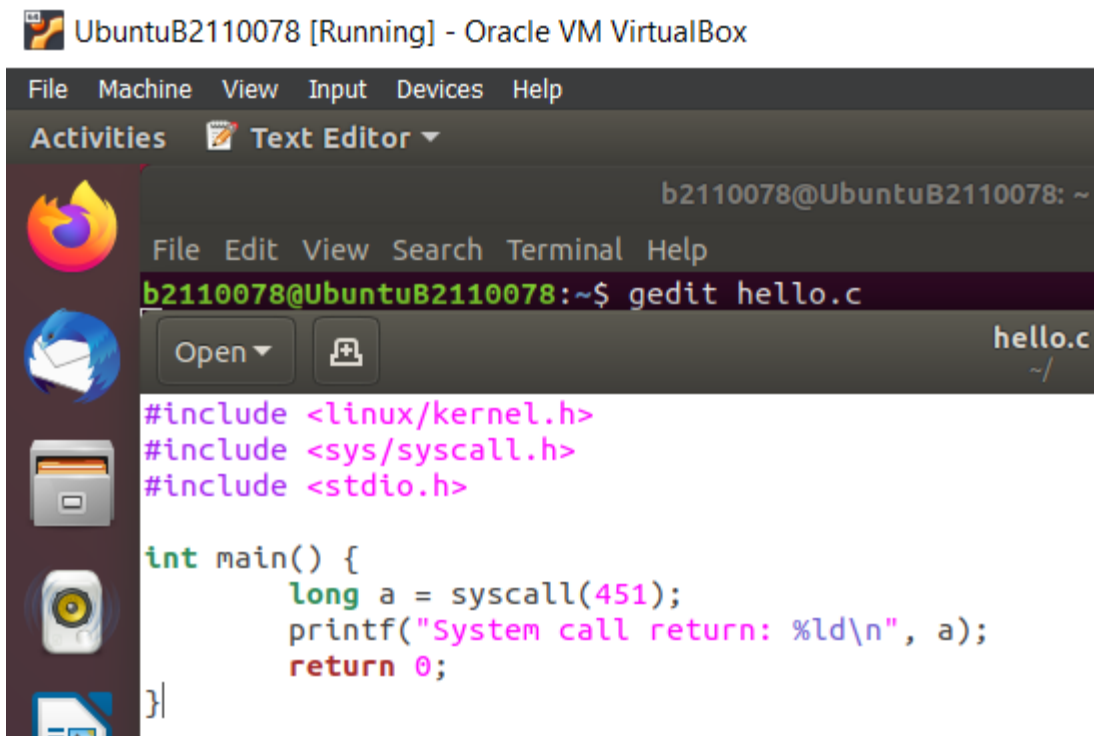
```
#include <sys/syscall.h>
```

```
#include <stdio.h>
```

```

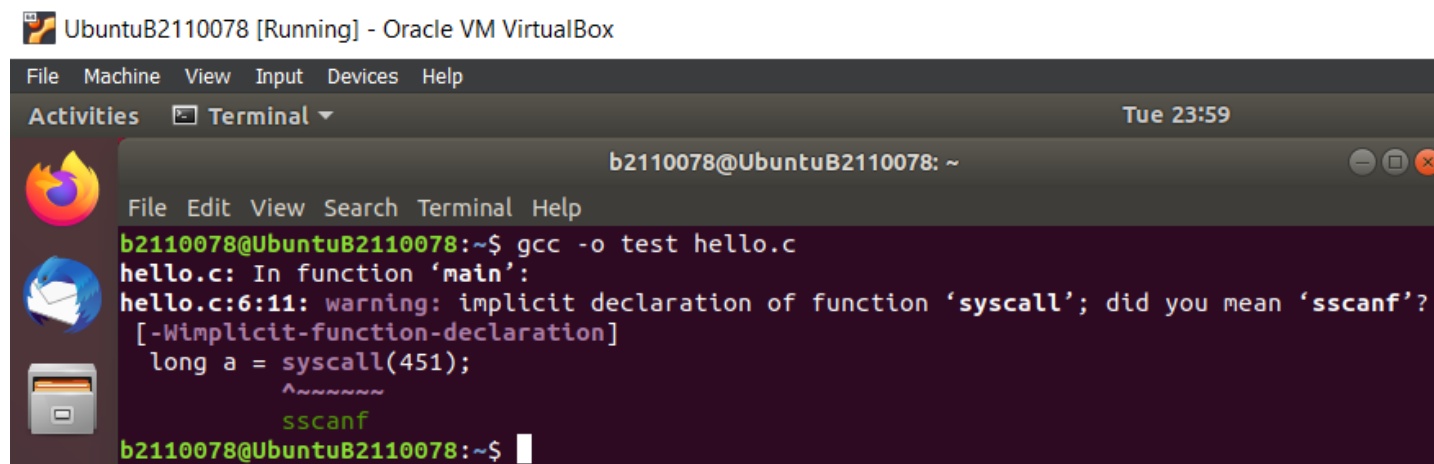
int main() {
    long a = syscall(451);
    printf("System call return: %ld\n", a);
    return 0;
}

```



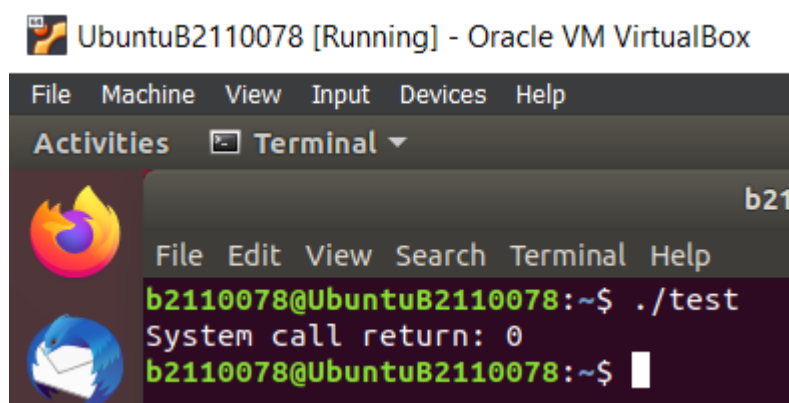
- Biên dịch chương trình hello.c và tạo file output test

# **gcc -o test hello.c**



- Thực hiện file test

# **./test**



- Nếu giá trị trở về là 0 thì lời gọi hệ thống đã được thực thi thành công.
- Khi xem log của kernel, ta có thể thấy câu “Xin chào. Ten toi la Quach Minh Hon – B2110078”

# **sudo dmesg**

```

root@UbuntuB2110078: /home/b2110078# sudo dmesg
[ 0.000000] Linux version 5.19.5 (root@UbuntuB2110078) (gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0, GNU ld (GNU Binutils for Ubuntu) 2.30) #1 SMP PREEMPT_DYNAMIC Tue Oct 10 21:21:47 +07 2023
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.19.5 root=UUID=0fa0f2d1-81dc-44fb-a850-9748a0269fad ro quiet splash
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
[ 0.000000] AMD AuthenticAMD
[ 0.000000] Hygon HygonGenuine
[ 0.000000] Centaur CentaurHauls
[ 0.000000] zhaoxin Shanghai
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[ 0.000000] signal: max sigframe size: 1776
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000100000-0x000000000000dffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000dfff0000-0x00000000000dfffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x000000000fec00000-0x000000000fec00fff] reserved
[ 0.000000] ...
[ 1574.240190] Xin chào. Ten toi la Quach Minh Hon - B2110078
root@UbuntuB2110078: /home/b2110078#

```

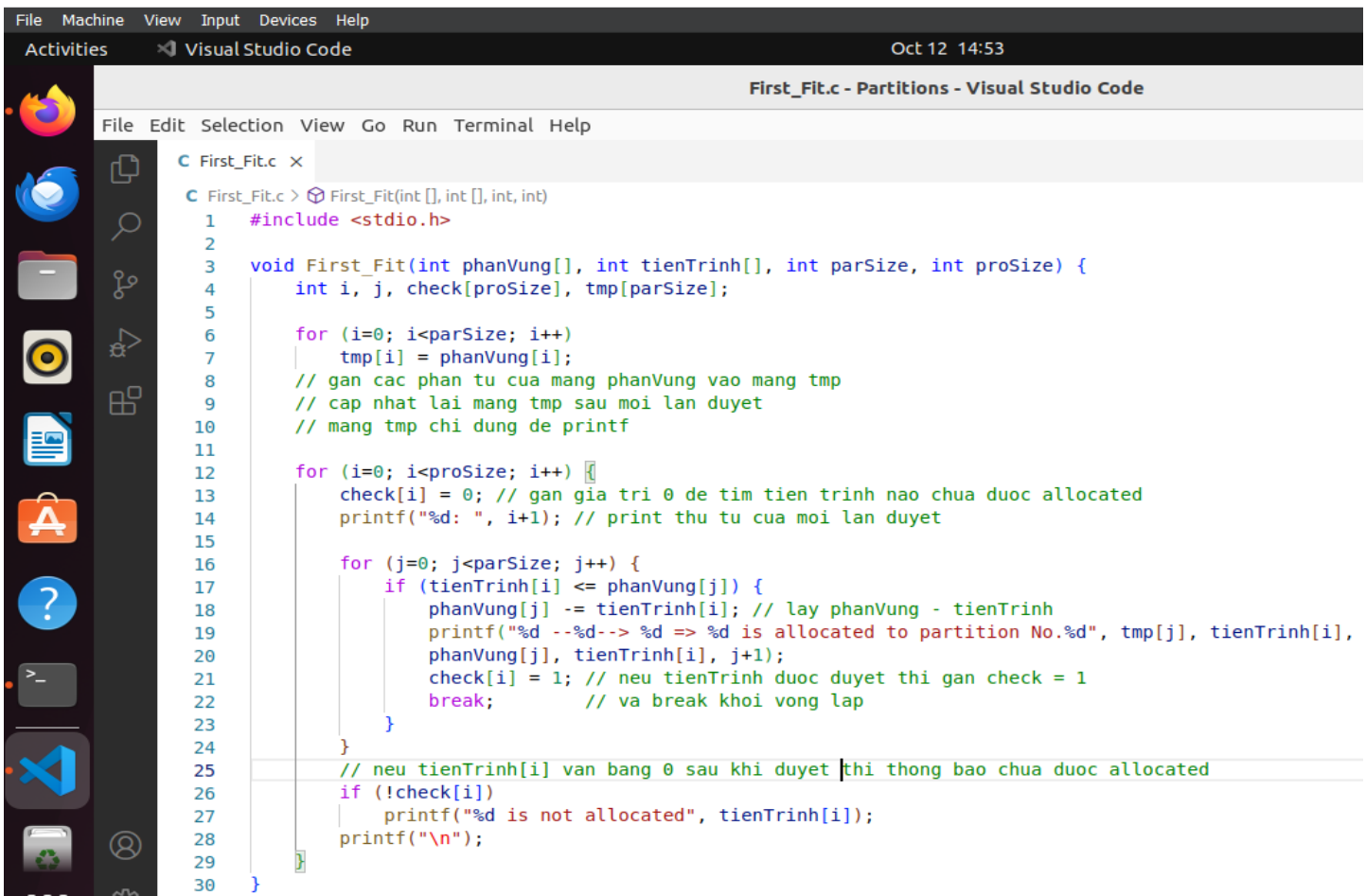
### PHẦN 3: Cấp phát vùng nhớ

Anh chị hãy cài đặt 3 thuật toán cấp phát vùng nhớ (First-fit, Best-Fit, Worst-Fit) cho tiến trình, đưa ví dụ cụ thể để test kết quả thực hiện.

- **Thuật toán First-fit:**



UbuntuB2110078 [Running] - Oracle VM VirtualBox



```

File Machine View Input Devices Help
Activities Visual Studio Code Oct 12 14:53

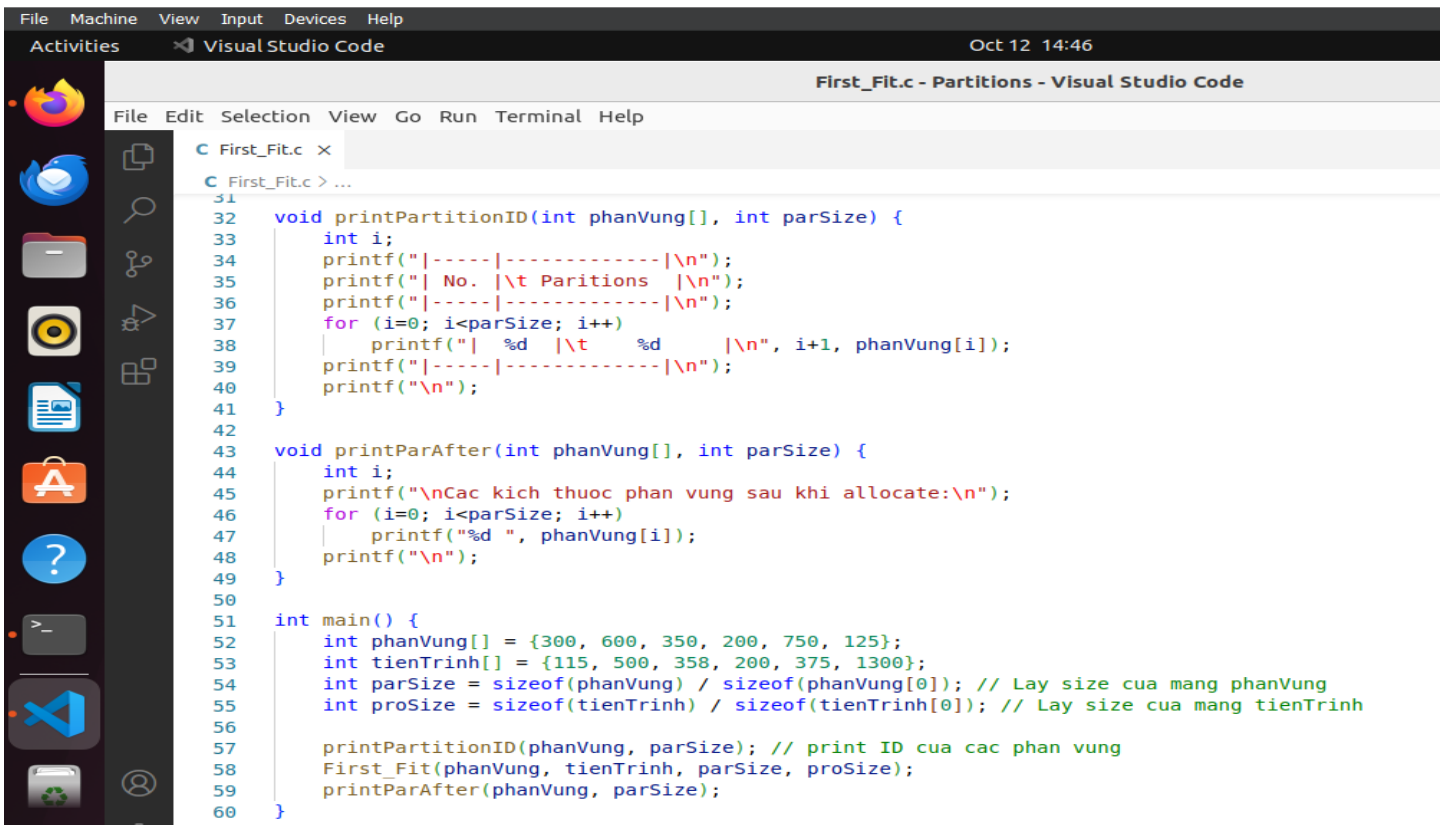
First_Fit.c - Partitions - Visual Studio Code

File Edit Selection View Go Run Terminal Help

C First_Fit.c x
C First_Fit.c > First_Fit(int [], int [], int, int)
1  #include <stdio.h>
2
3  void First_Fit(int phanVung[], int tienTrinh[], int parSize, int proSize) {
4      int i, j, check[proSize], tmp[parSize];
5
6      for (i=0; i<parSize; i++)
7          tmp[i] = phanVung[i];
8      // gan cac phan tu cua mang phanVung vao mang tmp
9      // cap nhat lai mang tmp sau moi lan duyet
10     // mang tmp chi dung de printf
11
12     for (i=0; i<proSize; i++) {
13         check[i] = 0; // gan gia tri 0 de tim tien trinh nao chua duoc allocated
14         printf("%d: ", i+1); // print thu tu cua moi lan duyet
15
16         for (j=0; j<parSize; j++) {
17             if (tienTrinh[i] <= phanVung[j]) {
18                 phanVung[j] -= tienTrinh[i]; // lay phanVung - tienTrinh
19                 printf("%d --%d--> %d => %d is allocated to partition No.%d", tmp[j], tienTrinh[i],
20                     phanVung[j], tienTrinh[i], j+1);
21                 check[i] = 1; // neu tienTrinh duoc duyet thi gan check = 1
22                 break; // va break khoi vong lap
23             }
24         }
25         // neu tienTrinh[i] van bang 0 sau khi duyet thi thong bao chua duoc allocated
26         if (!check[i])
27             printf("%d is not allocated", tienTrinh[i]);
28         printf("\n");
29     }
30 }

```

UbuntuB2110078 [Running] - Oracle VM VirtualBox



```

File Machine View Input Devices Help
Activities Visual Studio Code Oct 12 14:46

First_Fit.c - Partitions - Visual Studio Code

File Edit Selection View Go Run Terminal Help

C First_Fit.c x
C First_Fit.c > ...
31
32 void printPartitionID(int phanVung[], int parSize) {
33     int i;
34     printf("|-----|-----|\n");
35     printf("| No. | \t Partitions | \n");
36     printf("|-----|-----|\n");
37     for (i=0; i<parSize; i++)
38         printf("| %d | \t %d | \n", i+1, phanVung[i]);
39     printf("|-----|-----|\n");
40     printf("\n");
41 }
42
43 void printParAfter(int phanVung[], int parSize) {
44     int i;
45     printf("\nCac kich thuoc phan vung sau khi allocate:\n");
46     for (i=0; i<parSize; i++)
47         printf("%d ", phanVung[i]);
48     printf("\n");
49 }
50
51 int main() {
52     int phanVung[] = {300, 600, 350, 200, 750, 125};
53     int tienTrinh[] = {115, 500, 358, 200, 375, 1300};
54     int parSize = sizeof(phanVung) / sizeof(phanVung[0]); // Lay size cua mang phanVung
55     int proSize = sizeof(tienTrinh) / sizeof(tienTrinh[0]); // Lay size cua mang tienTrinh
56
57     printPartitionID(phanVung, parSize); // print ID cua cac phan vung
58     First_Fit(phanVung, tienTrinh, parSize, proSize);
59     printParAfter(phanVung, parSize);
60 }

```

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities Terminal

b2110078@UbuntuB2110078: ~/P
b2110078@UbuntuB2110078:~/Partitions$ gcc -o firstFit First_Fit.c
b2110078@UbuntuB2110078:~/Partitions$ ./firstFit

|-----|-----|
| No. | Paritions |
|-----|-----|
| 1 | 300 |
| 2 | 600 |
| 3 | 350 |
| 4 | 200 |
| 5 | 750 |
| 6 | 125 |
|-----|-----|

1: 300 --115--> 185 => 115 is allocated to partition No.1
2: 600 --500--> 100 => 500 is allocated to partition No.2
3: 750 --358--> 392 => 358 is allocated to partition No.5
4: 350 --200--> 150 => 200 is allocated to partition No.3
5: 750 --375--> 17 => 375 is allocated to partition No.5
6: 1300 is not allocated

Cac kích thước phân vùng sau khi allocate:
185 100 150 200 17 125
b2110078@UbuntuB2110078:~/Partitions$
```

### - Thuật toán Best-fit

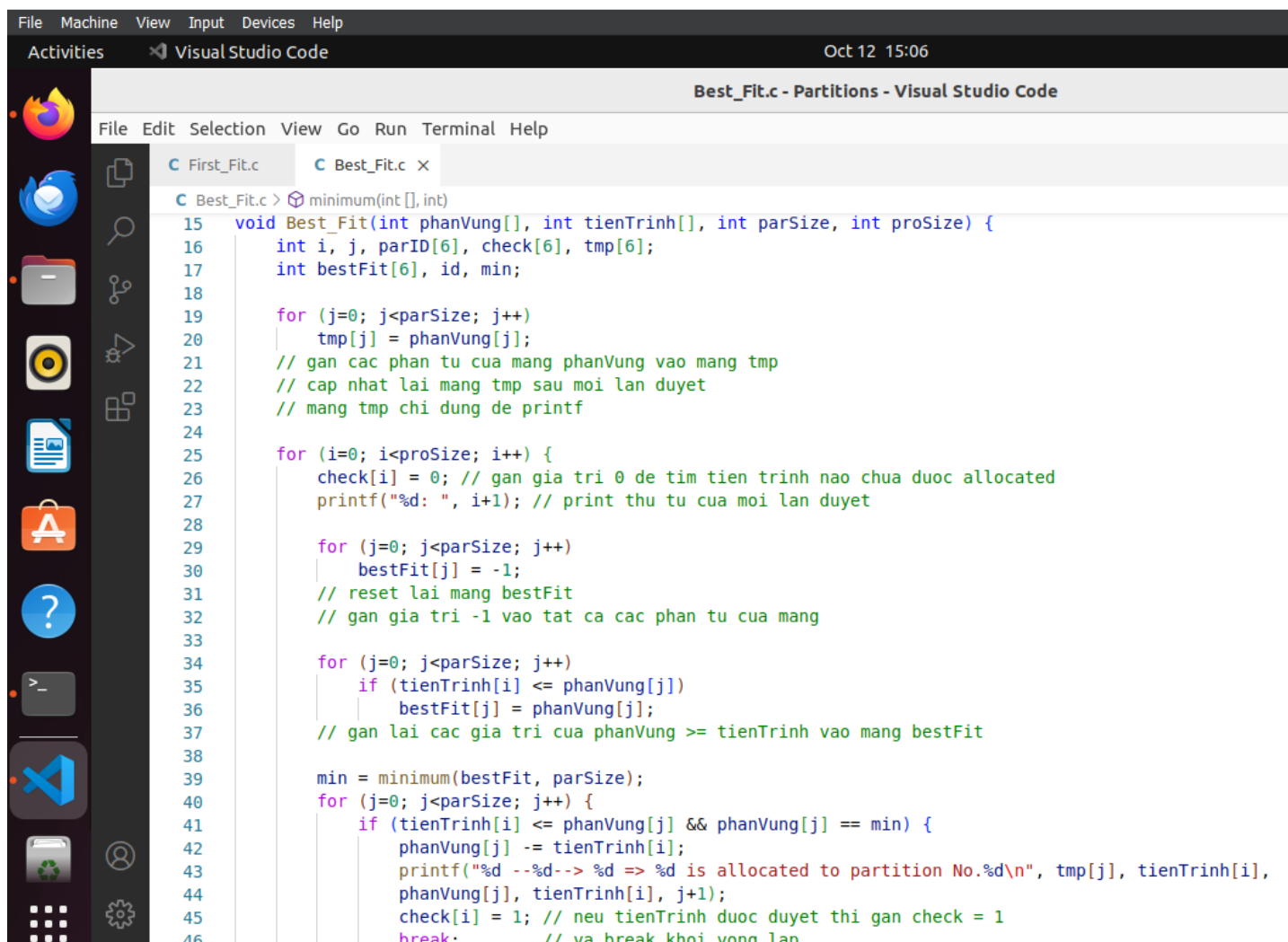
UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities Visual Studio Code Oct 12 15

Best_Fit.c - Partitions - V

File Edit Selection View Go Run Terminal Help


C First_Fit.c C Best_Fit.c x
C Best_Fit.c > minimum(int [], int)
1 #include <stdio.h>
2 #include <limits.h>
3
4 int minimum(int bestFit[], int parSize) {
5     int i, min=INT_MAX;
6     // gan min = gia tri max cua integer
7     for (i=1; i<parSize; i++)
8         if (bestFit[i] < min && bestFit[i] != -1)
9             min = bestFit[i];
10    // tim gia tri nho nhat nhung lon hon -1
11    // neu khong tim duoc gia tri nho nhat thi tra ve INT_MAX
12    return min;
13
14
```

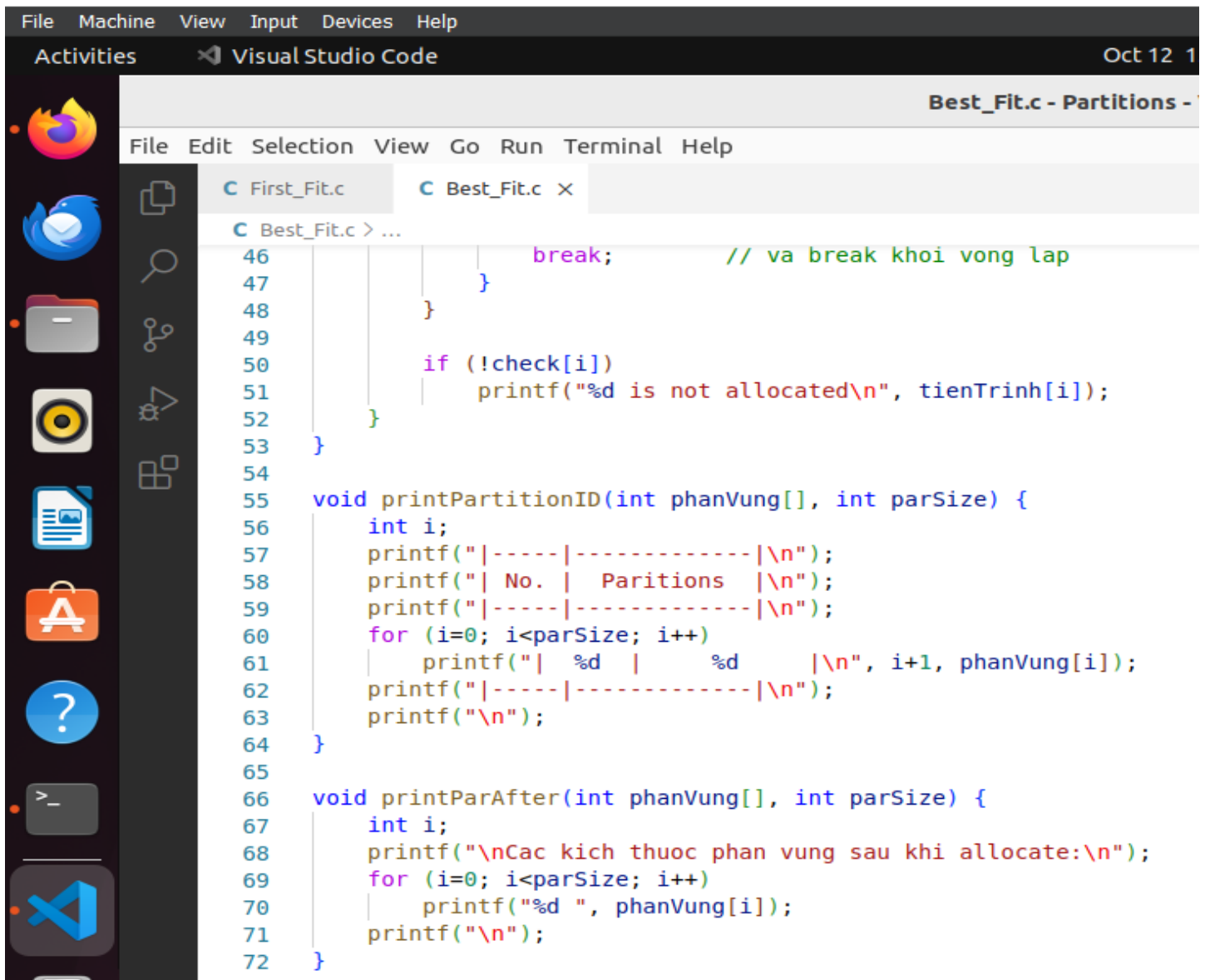


```

File Machine View Input Devices Help
Activities Visual Studio Code Oct 12 15:06
Best_Fit.c - Partitions - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C First_Fit.c C Best_Fit.c X
C Best_Fit.c > minimum(int [], int)
15 void Best_Fit(int phanVung[], int tienTrinh[], int parSize, int proSize) {
16     int i, j, parID[6], check[6], tmp[6];
17     int bestFit[6], id, min;
18
19     for (j=0; j<parSize; j++)
20         tmp[j] = phanVung[j];
21     // gan cac phan tu cua mang phanVung vao mang tmp
22     // cap nhat lai mang tmp sau moi lan duyet
23     // mang tmp chi dung de printf
24
25     for (i=0; i<proSize; i++) {
26         check[i] = 0; // gan gia tri 0 de tim tien trinh nao chua duoc allocated
27         printf("%d: ", i+1); // print thu tu cua moi lan duyet
28
29         for (j=0; j<parSize; j++)
30             bestFit[j] = -1;
31         // reset lai mang bestFit
32         // gan gia tri -1 vao tat ca cac phan tu cua mang
33
34         for (j=0; j<parSize; j++)
35             if (tienTrinh[i] <= phanVung[j])
36                 bestFit[j] = phanVung[j];
37         // gan lai cac gia tri cua phanVung >= tienTrinh vao mang bestFit
38
39         min = minimum(bestFit, parSize);
40         for (j=0; j<parSize; j++) {
41             if (tienTrinh[i] <= phanVung[j] && phanVung[j] == min) {
42                 phanVung[j] -= tienTrinh[i];
43                 printf("%d --%d--> %d => %d is allocated to partition No.%d\n", tmp[j], tienTrinh[i],
44                     phanVung[j], tienTrinh[i], j+1);
45                 check[i] = 1; // neu tienTrinh duoc duyet thi gan check = 1
46                 break; // va break khoi vong lan

```

 UbuntuB2110078 [Running] - Oracle VM VirtualBox



```

File Machine View Input Devices Help
Activities Visual Studio Code Oct 12 1


Best_Fit.c - Partitions -

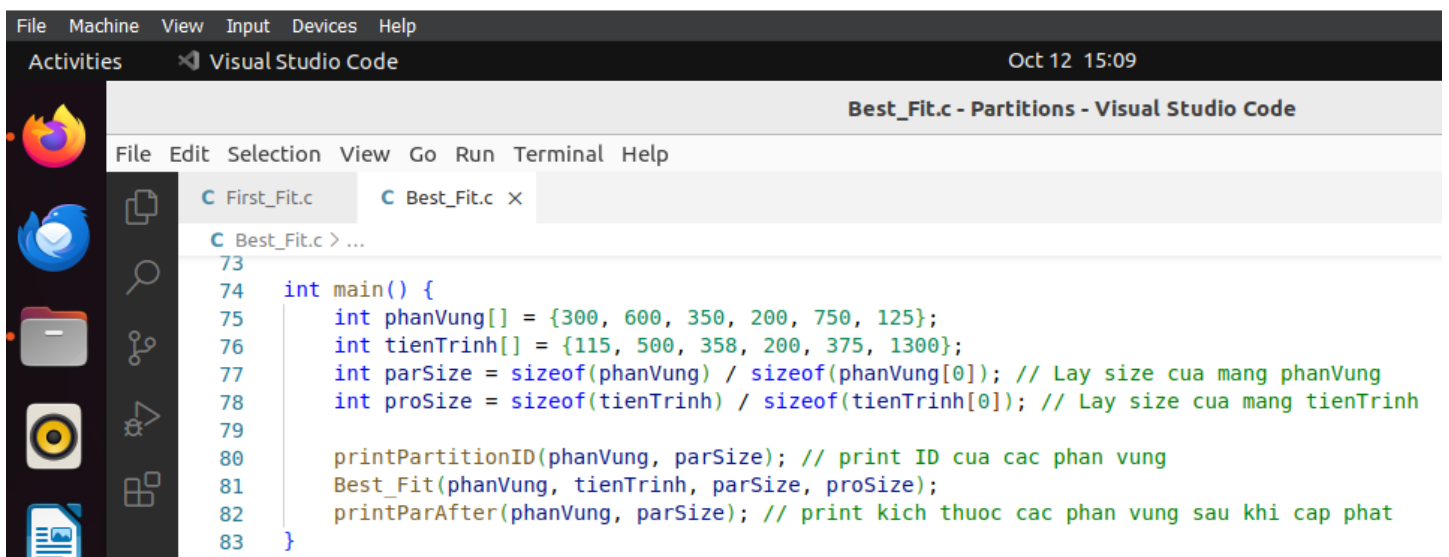
File Edit Selection View Go Run Terminal Help

C First_Fit.c C Best_Fit.c X

C Best_Fit.c > ...
46         break;           // va break khoi vong lap
47     }
48 }
49
50     if (!check[i])
51         printf("%d is not allocated\n", tienTrinh[i]);
52 }
53 }
54
55 void printPartitionID(int phanVung[], int parSize) {
56     int i;
57     printf("|-----|-----|\n");
58     printf("| No. | Partitions |\n");
59     printf("|-----|-----|\n");
60     for (i=0; i<parSize; i++)
61         printf("| %d | %d |\n", i+1, phanVung[i]);
62     printf("|-----|-----|\n");
63     printf("\n");
64 }
65
66 void printParAfter(int phanVung[], int parSize) {
67     int i;
68     printf("\nCac kích thước phân vùng sau khi allocate:\n");
69     for (i=0; i<parSize; i++)
70         printf("%d ", phanVung[i]);
71     printf("\n");
72 }
73
74 int main() {
75     int phanVung[] = {300, 600, 350, 200, 750, 125};
76     int tienTrinh[] = {115, 500, 358, 200, 375, 1300};
77     int parSize = sizeof(phanVung) / sizeof(phanVung[0]); // Lay size của mảng phanVung
78     int proSize = sizeof(tienTrinh) / sizeof(tienTrinh[0]); // Lay size của mảng tienTrinh
79
80     printPartitionID(phanVung, parSize); // print ID của các phân vùng
81     Best_Fit(phanVung, tienTrinh, parSize, proSize);
82     printParAfter(phanVung, parSize); // print kích thước các phân vùng sau khi cấp phát
83 }

```

 UbuntuB2110078 [Running] - Oracle VM VirtualBox



```

File Machine View Input Devices Help
Activities Visual Studio Code Oct 12 15:09

Best_Fit.c - Partitions - Visual Studio Code

File Edit Selection View Go Run Terminal Help

C First_Fit.c C Best_Fit.c X

C Best_Fit.c > ...
73
74 int main() {
75     int phanVung[] = {300, 600, 350, 200, 750, 125};
76     int tienTrinh[] = {115, 500, 358, 200, 375, 1300};
77     int parSize = sizeof(phanVung) / sizeof(phanVung[0]); // Lay size của mảng phanVung
78     int proSize = sizeof(tienTrinh) / sizeof(tienTrinh[0]); // Lay size của mảng tienTrinh
79
80     printPartitionID(phanVung, parSize); // print ID của các phân vùng
81     Best_Fit(phanVung, tienTrinh, parSize, proSize);
82     printParAfter(phanVung, parSize); // print kích thước các phân vùng sau khi cấp phát
83 }

```

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities Terminal

b2110078@UbuntuB2110078: ~/Partitions$ gcc -o bestFit Best_Fit.c
b2110078@UbuntuB2110078: ~/Partitions$ ./bestFit
```

No.	Partitions
1	300
2	600
3	350
4	200
5	750
6	125

```

1: 125 --115--> 10 => 115 is allocated to partition No.6
2: 600 --500--> 100 => 500 is allocated to partition No.2
3: 750 --358--> 392 => 358 is allocated to partition No.5
4: 200 --200--> 0 => 200 is allocated to partition No.4
5: 750 --375--> 17 => 375 is allocated to partition No.5
6: 1300 is not allocated

Cac kích thước phân vùng sau khi allocate:
300 100 350 0 17 10

```

### - Thuật toán Worst-fit

UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities Visual Studio Code

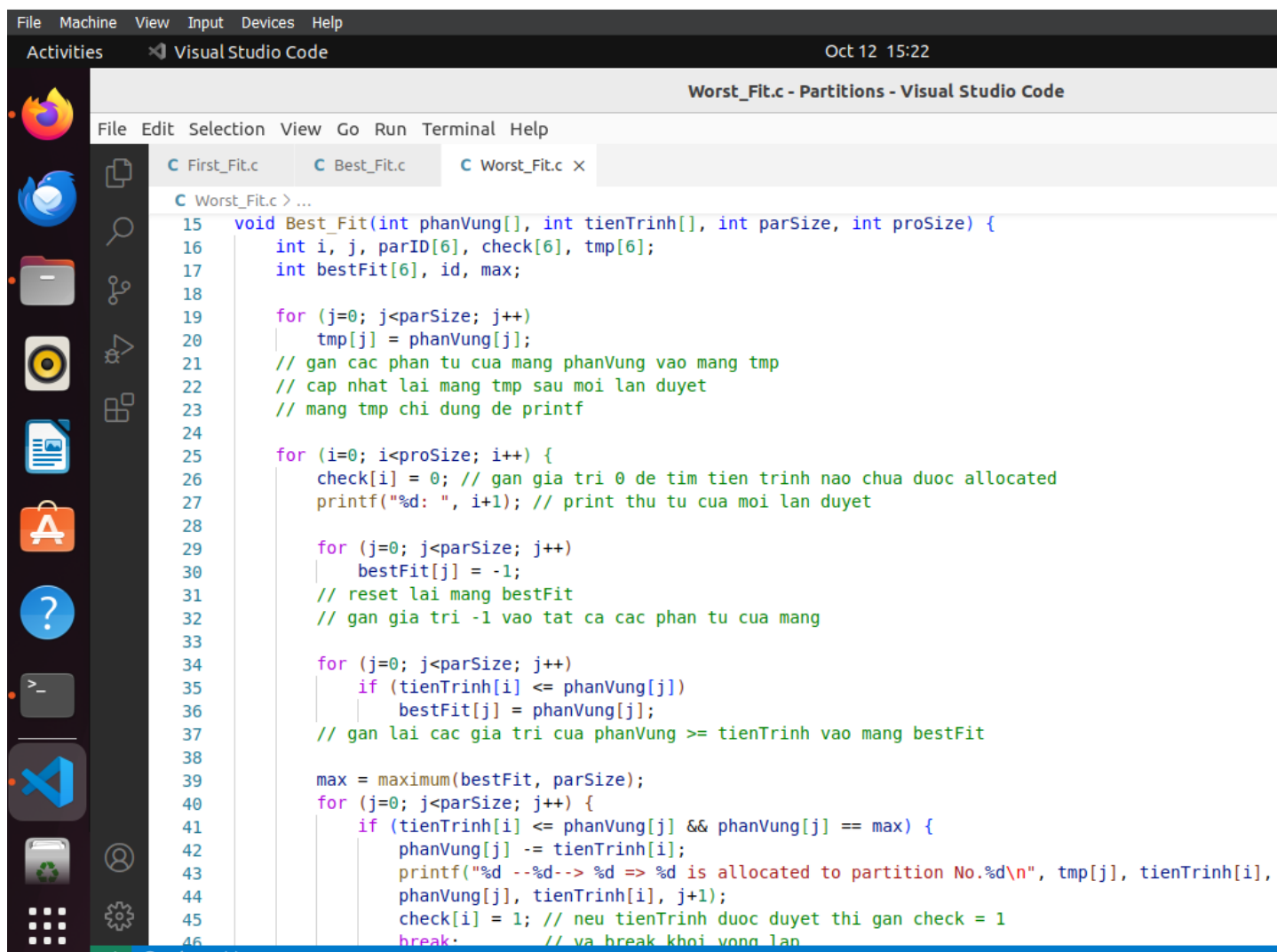
Worst_Fit.c - Partition

File Edit Selection View Go Run Terminal Help


C First_Fit.c C Best_Fit.c C Worst_Fit.c X

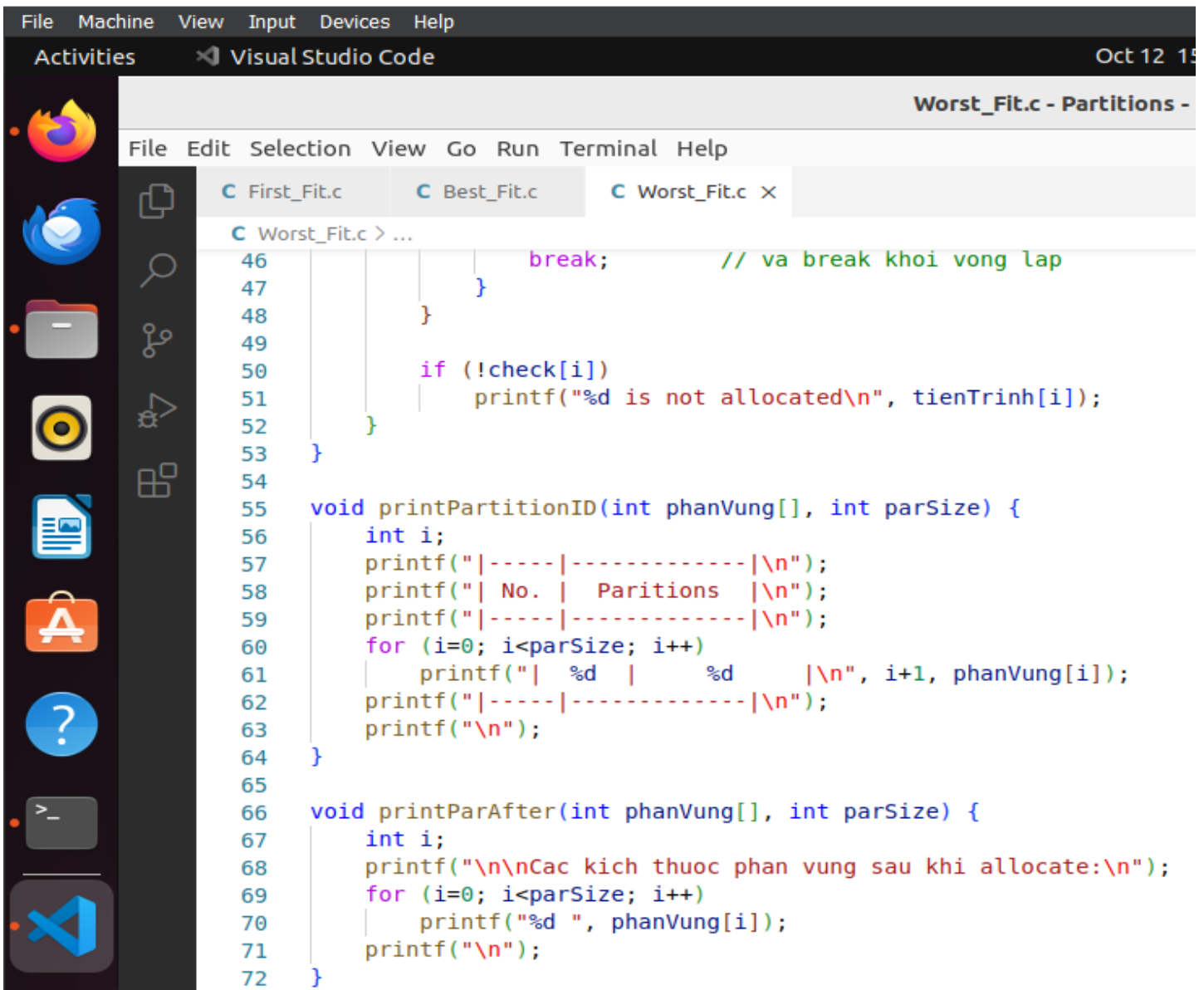
C Worst_Fit.c > ...
1  #include <stdio.h>
2  #include <limits.h>
3
4  int maximum(int bestFit[], int parSize) {
5      int i, max=INT_MIN;
6      // gan max = gia tri max cua integer
7      for (i=1; i<parSize; i++)
8          if (bestFit[i] > max && bestFit[i] != -1)
9              max = bestFit[i];
10     // tim gia tri nho nhat nhung lon hon -1
11     // neu khong tim duoc gia tri nho nhat thi tra ve INT_MAX
12     return max;
13 }
```





```
File Machine View Input Devices Help
Activities Visual Studio Code Oct 12 15:22
Worst_Fit.c - Partitions - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C First_Fit.c C Best_Fit.c C Worst_Fit.c x
C Worst_Fit.c > ...
15 void Best_Fit(int phanVung[], int tienTrinh[], int parSize, int proSize) {
16     int i, j, parID[6], check[6], tmp[6];
17     int bestFit[6], id, max;
18
19     for (j=0; j<parSize; j++)
20         tmp[j] = phanVung[j];
21     // gan cac phan tu cua mang phanVung vao mang tmp
22     // cap nhat lai mang tmp sau moi lan duyet
23     // mang tmp chi dung de printf
24
25     for (i=0; i<proSize; i++) {
26         check[i] = 0; // gan gia tri 0 de tim tien trinh nao chua duoc allocated
27         printf("%d: ", i+1); // print thu tu cua moi lan duyet
28
29         for (j=0; j<parSize; j++)
30             bestFit[j] = -1;
31         // reset lai mang bestFit
32         // gan gia tri -1 vao tat ca cac phan tu cua mang
33
34         for (j=0; j<parSize; j++)
35             if (tienTrinh[i] <= phanVung[j])
36                 bestFit[j] = phanVung[j];
37         // gan lai cac gia tri cua phanVung >= tienTrinh vao mang bestFit
38
39         max = maximum(bestFit, parSize);
40         for (j=0; j<parSize; j++) {
41             if (tienTrinh[i] <= phanVung[j] && phanVung[j] == max) {
42                 phanVung[j] -= tienTrinh[i];
43                 printf("%d --%d--> %d => %d is allocated to partition No.%d\n", tmp[j], tienTrinh[i],
44                     phanVung[j], tienTrinh[i], j+1);
45                 check[i] = 1; // neu tienTrinh duoc duyet thi gan check = 1
46                 break; // va break khoi vong lan
```


 UbuntuB2110078 [Running] - Oracle VM VirtualBox

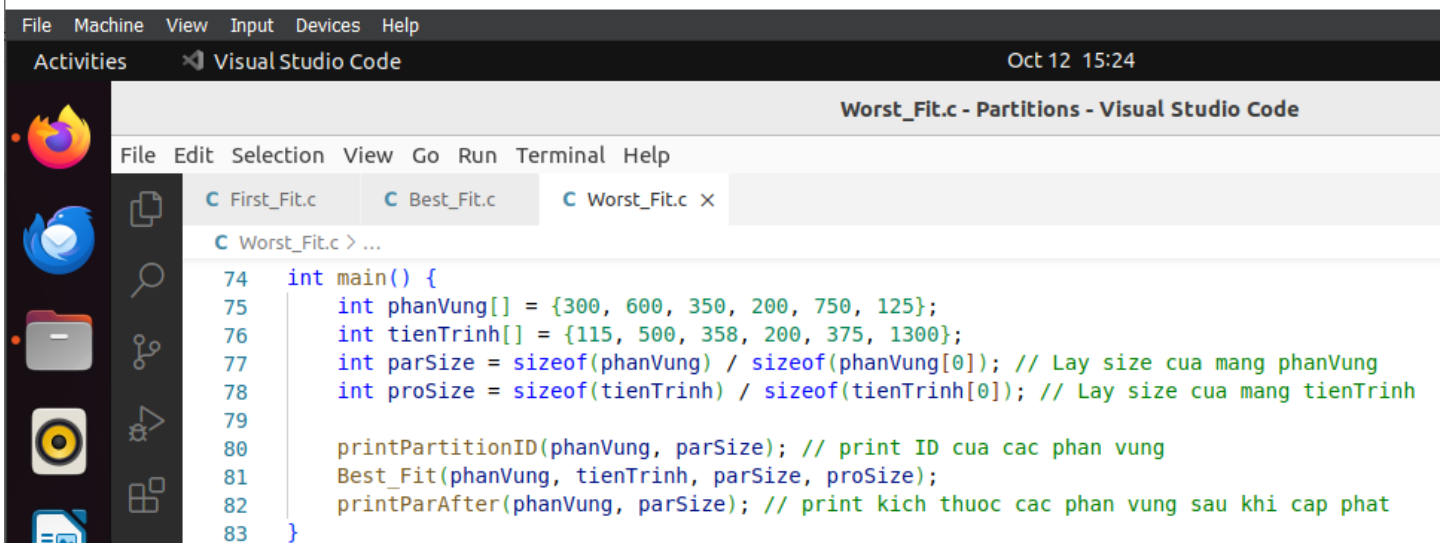


```

46         break;           // va break khoi vong lap
47     }
48 }
49
50     if (!check[i])
51         printf("%d is not allocated\n", tienTrinh[i]);
52 }
53 }
54
55 void printPartitionID(int phanVung[], int parSize) {
56     int i;
57     printf("|-----|-----|\n");
58     printf("| No. | Partitions |\n");
59     printf("|-----|-----|\n");
60     for (i=0; i<parSize; i++)
61         printf("| %d | %d |\n", i+1, phanVung[i]);
62     printf("|-----|-----|\n");
63     printf("\n");
64 }
65
66 void printParAfter(int phanVung[], int parSize) {
67     int i;
68     printf("\n\nCac kích thước phân vùng sau khi allocate:\n");
69     for (i=0; i<parSize; i++)
70         printf("%d ", phanVung[i]);
71     printf("\n");
72 }

```


 UbuntuB2110078 [Running] - Oracle VM VirtualBox




```

74 int main() {
75     int phanVung[] = {300, 600, 350, 200, 750, 125};
76     int tienTrinh[] = {115, 500, 358, 200, 375, 1300};
77     int parSize = sizeof(phanVung) / sizeof(phanVung[0]); // Lay size của mảng phanVung
78     int proSize = sizeof(tienTrinh) / sizeof(tienTrinh[0]); // Lay size của mảng tienTrinh
79
80     printPartitionID(phanVung, parSize); // print ID của các phân vùng
81     Best_Fit(phanVung, tienTrinh, parSize, proSize);
82     printParAfter(phanVung, parSize); // print kích thước các phân vùng sau khi cấp phát
83 }

```

 UbuntuB2110078 [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Activities  Terminal

b2110078@UbuntuB2110078: ~/Partitions
b2110078@UbuntuB2110078:~/Partitions$ gcc -o worstFit Worst_Fit.c
b2110078@UbuntuB2110078:~/Partitions$ ./worstFit
```

No.	Partitions
1	300
2	600
3	350
4	200
5	750
6	125

```

1: 750 --115--> 635 => 115 is allocated to partition No.5
2: 750 --500--> 135 => 500 is allocated to partition No.5
3: 600 --358--> 242 => 358 is allocated to partition No.2
4: 350 --200--> 150 => 200 is allocated to partition No.3
5: 375 is not allocated
6: 1300 is not allocated

Cac kích thước phân vùng sau khi allocate:
300 242 150 200 135 125

```

\*\*\*\*\*HẾT\*\*\*\*\*