

## PHẦN 3: LÀM VIỆC VỚI TIẾN TRÌNH

\*Phiên bản Ubuntu sử dụng hiện tại: 16.04.6

\*Nên cấp cho máy số nhân  $\geq 2$  để máy biên dịch nhanh hơn

\*Thực hiện lệnh ở chế độ người dùng root (`$sudo -s`)

### 1. Xác định phiên bản hiện tại của kernel:

`uname -r`

```
cod3r@cod3r-VirtualBox:~$ uname -r
4.15.0-45-generic
```

### 2. Tải về source của kernel:

`wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.17.4.tar.xz`

Trong project này, em tải về kernel có phiên bản cao hơn kernel hiện tại (4.17.4). Mục đích là sau khi biên dịch và khởi động lại chương trình, kernel sẽ tự động cập nhật.

```
root@cod3r-VirtualBox:~# wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.17.4.tar.xz
--2020-03-11 17:06:19-- https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.17.4.tar.xz
Resolving www.kernel.org (www.kernel.org)... 147.75.46.191, 2604:1380:4080:c00::1
Connecting to www.kernel.org (www.kernel.org)|147.75.46.191|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.17.4.tar.xz [following]
--2020-03-11 17:06:20-- https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.17.4.tar.xz
Resolving mirrors.edge.kernel.org (mirrors.edge.kernel.org)... 147.75.95.133, 2604:1380:3000:1500::1
Connecting to mirrors.edge.kernel.org (mirrors.edge.kernel.org)|147.75.95.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 102176828 (97M) [application/x-xz]
Saving to: 'linux-4.17.4.tar.xz.1'

linux-4.17.4.tar.xz  7%[>                ]  7.69M  2.56MB/s  eta 35s
```

### 3. Giải nén source của kernel vừa tải về vào thư mục gốc của kernel:

`tar -xvf linux-4.17.4.tar.xz -C /usr/src/`

```
root@cod3r-VirtualBox:~# tar -xvf linux-4.17.4.tar.xz -C /usr/src/  
linux-4.17.4/  
linux-4.17.4/.clang-format  
linux-4.17.4/.cocciconfig  
linux-4.17.4/.get_maintainer.ignore  
linux-4.17.4/.gitattributes  
linux-4.17.4/.gitignore  
linux-4.17.4/.mailmap  
linux-4.17.4/COPYING  
linux-4.17.4/CREDITS  
linux-4.17.4/Documentation/  
linux-4.17.4/Documentation/.gitignore  
linux-4.17.4/Documentation/00-INDEX  
linux-4.17.4/Documentation/ABI/  
linux-4.17.4/Documentation/ABI/README  
linux-4.17.4/Documentation/ABI/obsolete/  
linux-4.17.4/Documentation/ABI/obsolete/sysfs-bus-usb  
linux-4.17.4/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-arvo  
linux-4.17.4/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-isku  
linux-4.17.4/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-koneplus  
linux-4.17.4/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-konepure  
linux-4.17.4/Documentation/ABI/obsolete/sysfs-driver-hid-roccat-kovaplus
```

#### 4. Định nghĩa một system call mới:

Em sẽ đặt tên cho system call mới là `sys_hello`. Lời gọi hệ thống này sẽ xác định tiến trình gọi (calling process) ở mức người dùng (user-level) và in ra Process id, trạng thái tiến trình (state) và tên chương trình (program name).

Di chuyển vào thư mục vừa mới giải nén xong:

```
cd /usr/src/linux-4.17.4/
```

Tạo một thư mục tên `hello` và di chuyển vào trong nó

```
mkdir hello
```

```
cd hello
```

```
root@cod3r-VirtualBox:/usr/src/linux-4.17.4# mkdir hello  
root@cod3r-VirtualBox:/usr/src/linux-4.17.4# cd hello  
root@cod3r-VirtualBox:/usr/src/linux-4.17.4/hello# gedit hello.c
```

Tạo một file `hello.c`

```
gedit hello.c
```

với nội dung như sau:

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

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

```
#include<linux/sched.h>
#include<linux/syscalls.h>
#include "hello.h"
asmlinkage long sys_hello(void) {
    struct task_struct *proces=current;
    for_each_process(proces) {
        printk(
            "Process: %s\n
            PID_Number: %ld\n
            Process State: %ld\n" ,
            proces->comm,
            (long)task_pid_nr(proces),
            (long)proces->state );
        printk("\n\n");
    }
    return 0;
}
```

```
#include<linux/kernel.h>
#include<linux/init.h>
#include<linux/sched.h>
#include<linux/syscalls.h>
#include "hello.h"
asmlinkage long sys_hello(void)
{
    struct task_struct *proces=current;

    for_each_process(proces) {

        printk(
            "Process: %s\n \
            PID_Number: %ld\n \
            Process State: %ld\n",

            proces->comm,
            (long)task_pid_nr(proces),
            (long) proces->state );
        printk("\n\n");
    }

    return 0;
}
```

Save lại và thoát

Tiếp tục tạo một file Makefile

*gedit Makefile*

với nội dung như sau:

*obj-y := hello.o*

---

```
obj-y := hello.o
```

Save lại và thoát

Cuối cùng tạo một file hello.h

```
cod3r@cod3r-VirtualBox:/usr/src/linux-4.17.4/hello$ gedit hello.h
```

với nội dung như sau:

asmlinkage long sys\_hello(void);

---

```
asmlinkage long sys_hello(void);
```

Save lại và thoát

## 5. Thêm tên của thư mục vừa tạo vào file Makefile của kernel:

Di chuyển ra thư mục /linux-4.17.4/ và chỉnh sửa nội dung của file Makefile:

```
root@cod3r-VirtualBox: /usr/src/linux-4.17.4/hello# cd ..
root@cod3r-VirtualBox: /usr/src/linux-4.17.4# gedit Makefile
```

Tìm dòng này `core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/`

Và thêm `hello/` vào cuối dòng `core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/ hello/`



The screenshot shows a text editor window with the Makefile content. The line `core-y += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ hello/` is highlighted in orange. The search bar at the top right shows 'core-y' and '2 of 6' matches. The Makefile content includes variables like `SKIP_STACK_VALIDATION`, `PHONY`, `ifeq`, `core-y`, `vmlinux-dirs`, `vmlinux-alldirs`, `init-y`, `core-y`, `drivers-y`, `net-y`, `libs-y1`, `libs-y2`, and `virt-y`.

Save lại và thoát.

## 6. Thêm system call mới vào bảng system call

Di chuyển vào địa chỉ sau

```
cd arch/x86/entry/syscalls/
```

```
gedit syscall_64.tbl
```

```
root@cod3r-VirtualBox: /usr/src/linux-4.17.4/arch/x86/entry/syscalls# gedit syscall_64.tbl
```

thêm vào dòng cuối cùng nội dung như sau

548	64	hello	sys_hello
541	x32	setsockopt	__x32_compat_sys_setsockopt
542	x32	getsockopt	__x32_compat_sys_getsockopt
543	x32	io_setup	__x32_compat_sys_io_setup
544	x32	io_submit	__x32_compat_sys_io_submit
545	x32	execveat	__x32_compat_sys_execveat/ptregs
546	x32	preadv2	__x32_compat_sys_preadv64v2
547	x32	pwritev2	__x32_compat_sys_pwritev64v2
548	64	hello	sys_hello

Ở đây em ghi 548 là bởi vì dòng trước có số thứ tự là 547.

Tiếp theo ghi số 64 là bởi vì máy em đang sử dụng là 64 bit

Save lại và thoát

## 7. Thêm system call mới vào file system call header

Di chuyển vào địa chỉ sau

```
cd include/linux/
gedit syscalls.h
```

```
root@cod3r-VirtualBox:/usr/src/linux-4.17.4# cd include/linux/
root@cod3r-VirtualBox:/usr/src/linux-4.17.4/include/linux# gedit syscalls.h
```

thêm vào dòng cuối cùng nội dung như sau

*asmlinkage long sys\_hello(void);*

```
extern long do_sys_truncate(const char __user *pathname, loff_t length);
static inline long ksys_truncate(const char __user *pathname, loff_t length)
{
    return do_sys_truncate(pathname, length);
}
static inline unsigned int ksys_personality(unsigned int personality)
{
    unsigned int old = current->personality;
    return old;
}
asmlinkage long sys_hello(void);
#endif
```

Save lại và thoát

## 8. Biên dịch kernel:

Trước khi biên dịch, em đã tải về một số gói hỗ trợ cũng như cập nhật nâng cấp cho chương trình của hệ thống:

```
apt-get install gcc libncurses5-dev bison flex libssl-dev libelf-dev
```



*apt-get update*

*apt-get upgrade*

```
root@cod3r-VirtualBox:~# apt-get install gcc libncurses5-dev libssl-dev libelf-dev bison flex
Reading package lists... Done
Building dependency tree
Reading state information... Done
bison is already the newest version (2:3.0.4.dfsg-1).
flex is already the newest version (2.6.0-11).
gcc is already the newest version (4:5.3.1-1ubuntu1).
libncurses5-dev is already the newest version (6.0+20160213-1ubuntu1).
libelf-dev is already the newest version (0.165-3ubuntu1.2).
libssl-dev is already the newest version (1.0.2g-1ubuntu4.15).
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
```

```
root@cod3r-VirtualBox:~# apt-get update
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:2 http://sg.archive.ubuntu.com/ubuntu xenial InRelease
Get:3 http://sg.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:4 http://sg.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:5 http://security.ubuntu.com/ubuntu xenial-security/main amd64 DEP-11 Metadata [74.8 kB]
Get:6 http://sg.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Metadata [322 kB]
Get:7 http://security.ubuntu.com/ubuntu xenial-security/main DEP-11 64x64 Icons [83.8 kB]
Get:8 http://sg.archive.ubuntu.com/ubuntu xenial-updates/main DEP-11 64x64 Icons [236 kB]
Get:9 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 Metadata [124 kB]
Get:10 http://security.ubuntu.com/ubuntu xenial-security/universe DEP-11 64x64 Icons [194 kB]
Get:11 http://sg.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 DEP-11
```

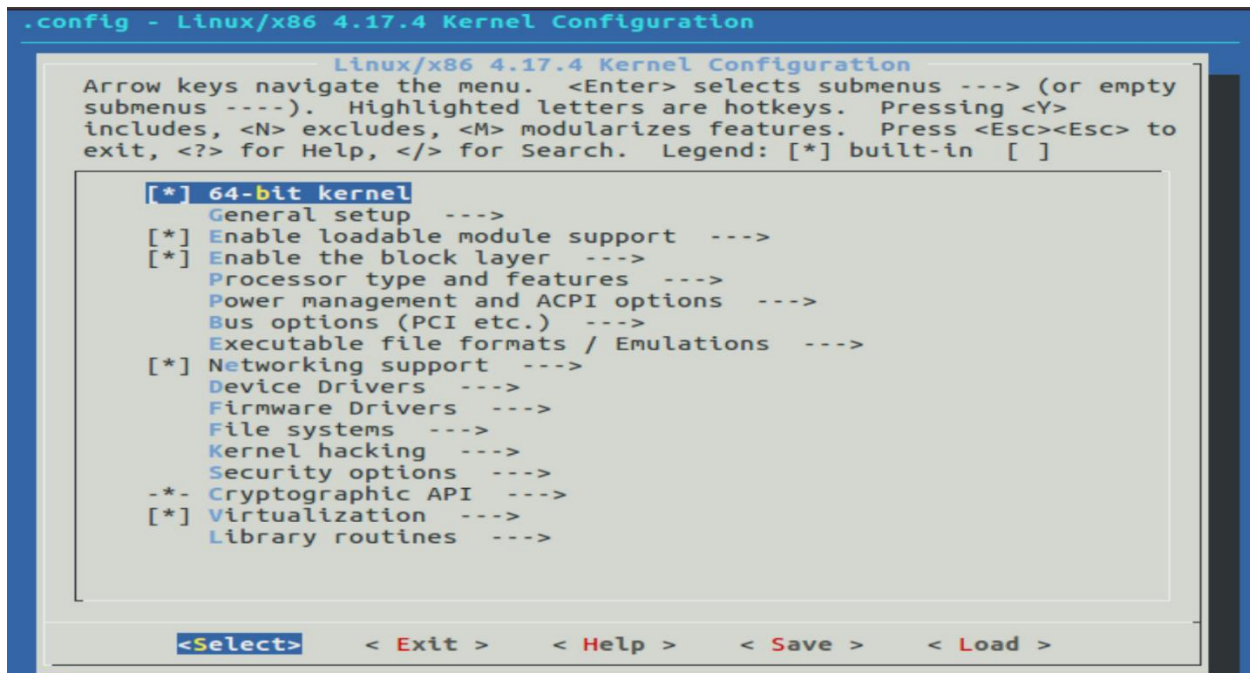
```

root@cod3r-VirtualBox:~# apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following packages have been kept back:
  linux-generic-hwe-16.04 linux-headers-generic-hwe-16.04
  linux-image-generic-hwe-16.04 ubuntu-desktop
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.

```

Tạo file cấu hình .config:

*make menuconfig*



Khi có cửa sổ hiện lên chọn Save chọn OK để tạo file .config

Chọn Exit 2 lần để thoát

Gõ lệnh nproc để xem chương trình linux đang sử dụng bao nhiêu nhân

Gõ lệnh *make -j số-nhân*

Em dùng lệnh này để giúp máy biên dịch nhanh hơn



```

root@cod3r-VirtualBox:/usr/src/linux-4.17.4# make -j 4
CHK      include/config/kernel.release
CHK      include/generated/uapi/linux/version.h
DESCEND  objtool
CHK      scripts/mod/devicetable-offsets.h
CHK      include/generated/utsrelease.h
CHK      include/generated/bounds.h
CHK      include/generated/timeconst.h
CHK      include/generated/asm-offsets.h
CALL     scripts/checksyscalls.sh
CHK      include/generated/compile.h

```

## 9. Cài đặt/ Cập nhật kernel:

Gõ lệnh *make modules\_install install*

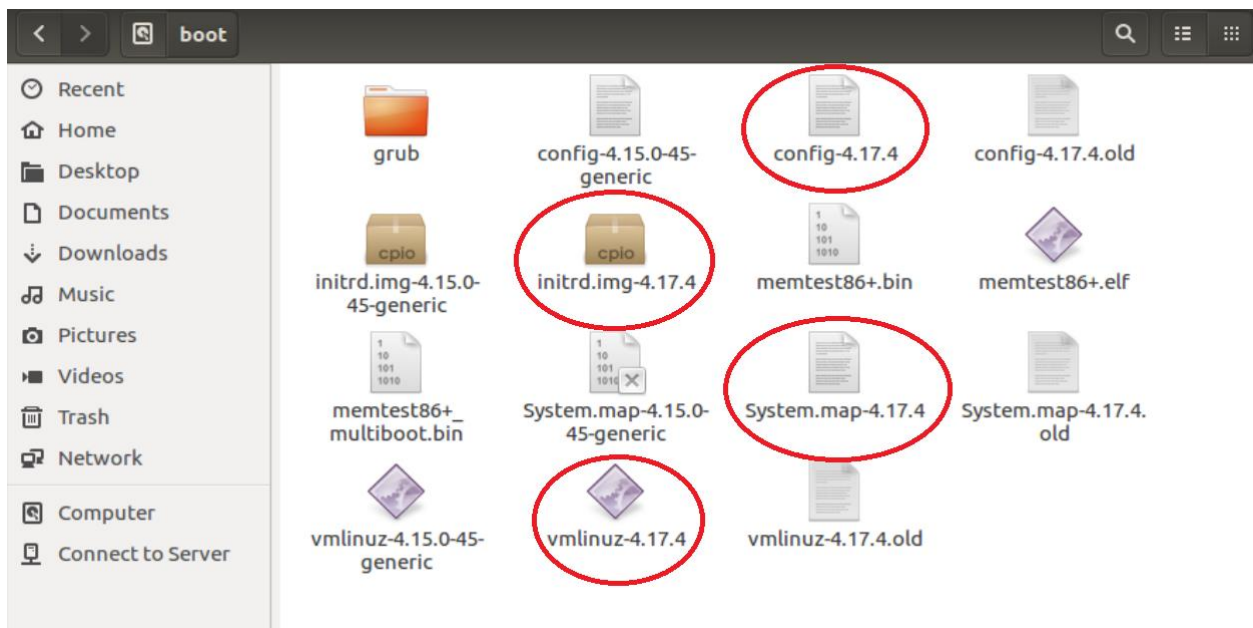
```

root@cod3r-VirtualBox:/usr/src/linux-4.17.4# make modules_install install
INSTALL arch/x86/crypto/aes-x86_64.ko
INSTALL arch/x86/crypto/aesni-intel.ko
INSTALL arch/x86/crypto/blowfish-x86_64.ko
INSTALL arch/x86/crypto/camellia-aesni-avx-x86_64.ko
INSTALL arch/x86/crypto/camellia-aesni-avx2.ko
INSTALL arch/x86/crypto/camellia-x86_64.ko
INSTALL arch/x86/crypto/cast5-avx-x86_64.ko
INSTALL arch/x86/crypto/cast6-avx-x86_64.ko
INSTALL arch/x86/crypto/chacha20-x86_64.ko
INSTALL arch/x86/crypto/crc32-pclmul.ko
INSTALL arch/x86/crypto/crct10dif-pclmul.ko
INSTALL arch/x86/crypto/des3_ede-x86_64.ko
INSTALL arch/x86/crypto/ghash-clmulni-intel.ko
INSTALL arch/x86/crypto/glue_helper.ko
INSTALL arch/x86/crypto/poly1305-x86_64.ko
INSTALL arch/x86/crypto/serpent-avx-x86_64.ko
INSTALL arch/x86/crypto/serpent-avx2.ko
INSTALL arch/x86/crypto/serpent-sse2-x86_64.ko
INSTALL arch/x86/crypto/sha1-mb/sha1-mb.ko
INSTALL arch/x86/crypto/sha1-ssse3.ko
INSTALL arch/x86/crypto/sha256-mb/sha256-mb.ko

```

Vào thư mục /boot kiểm tra xem có tồn tại các file dưới đây hay không. Nếu có tức là quá trình cài đặt đã hoàn thành

1. System.map-4.17.4
2. vmlinuz-4.17.4
3. initrd.img-4.17.4
4. config-4.17.4



Bây giờ chỉ cần khởi động lại bằng cách gõ lệnh *reboot* là chương trình sẽ tự động cập nhật kernel.

Sau khi khởi động lại em đã kiểm tra bằng lệnh *uname -r*

```
cod3r@cod3r-VirtualBox:~$ uname -r  
4.17.4
```

## 10. Kiểm tra system call

Đứng ở thư mục `/home` và tạo một file `userspace.c` với nội dung như sau:

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

```

#include <sys/syscall.h>
#include <unistd.h>
int main()
{
    printf("Invoking 'hello' system call...\n");

    long int ret_status = syscall(548); //548 là số thứ tự của syscall đã định
    nghĩa ở trên

    if(ret_status == 0)
        printf("System call 'hello' executed correctly. Use dmesg to check
        hello\n");

    else
        printf("System call 'hello' did not execute as expected\n");

    return 0;
}

```

---

```

#include <stdio.h>
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <unistd.h>
int main()
{
    printf("Invoking 'Hello' system call");

    long int ret_status = syscall(548);

    if(ret_status == 0)
        printf("System call 'Hello' executed correctly. Use dmesg to check Hello\n");

    else
        printf("System call 'Hello' did not execute as expected\n");

    return 0;
}

```

Tiếp theo biên dịch file userspace.c và chạy file kết quả

`gcc userspace.c`

`./a.out`

```

root@cod3r-VirtualBox:~# gcc userspace.c
root@cod3r-VirtualBox:~# ./a.out
Invoking 'Hello' system call...
System call 'Hello' executed correctly. Use dmesg to check Hello

```

File thông báo system call hello đã thực thi thành công. Gõ lệnh *dmesg* để check syslog

```
Process State: 1
[ 148.634539]
[ 148.634540] Process: deja-dup-monito
                PID_Number: 1907
                Process State: 1
[ 148.634540]
[ 148.634541] Process: sudo
                PID_Number: 1916
                Process State: 1
[ 148.634542]
[ 148.634543] Process: bash
                PID_Number: 1919
                Process State: 1
[ 148.634543]
[ 148.634544] Process: a.out
                PID_Number: 1940
                Process State: 0
[ 148.634545]
```