

Họ và tên: Ngô Hồng Quốc Bảo

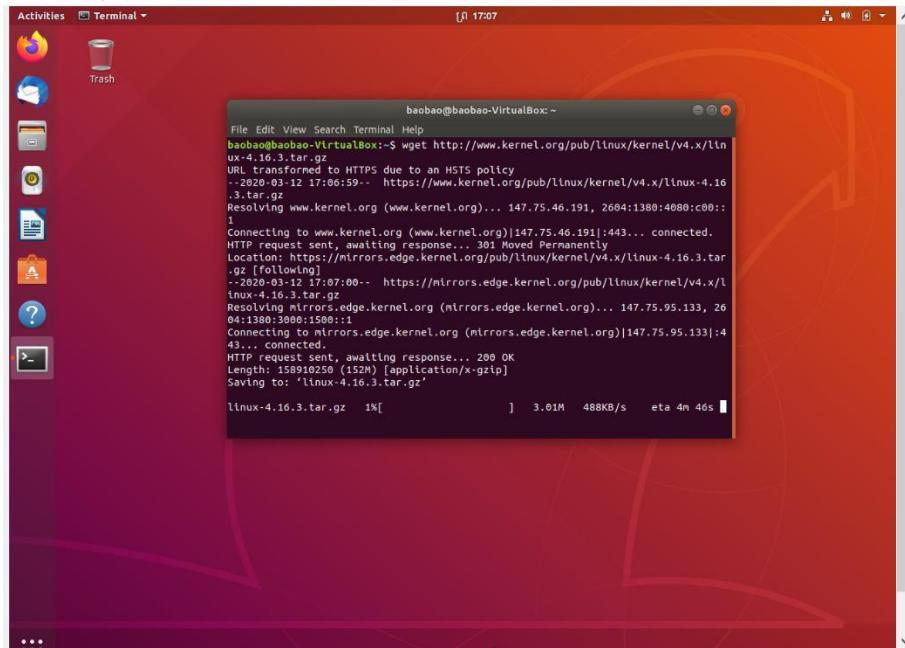
MSSV: B1809677

## PROJECT

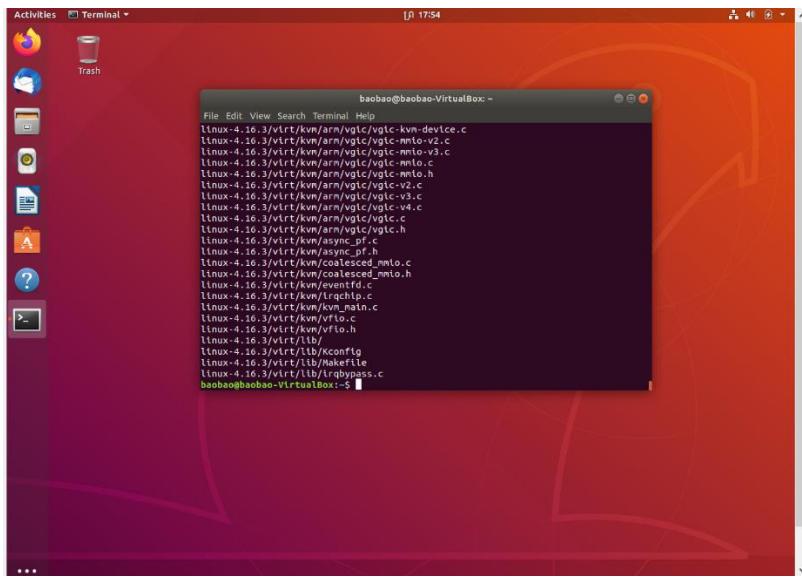
# Nguyên Lý Hệ Điều Hành

### Phần 1: Xây Dựng Linux Kernel

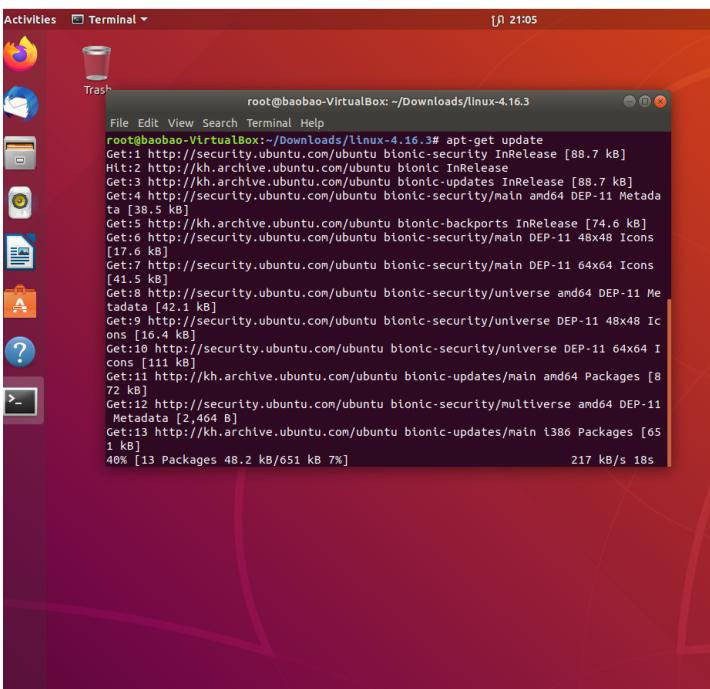
- sudo -s (bật chế độ root user)
- Tải các tool cần thiết
  - o apt-get install -y gcc libncurses5-dev make wget
  - o apt-get install -y gcc libssl-dev
  - o apt-get install bison flex
- Tải và giải nén file linux 4.16.3:
  - o wget <http://www.kernel.org/pub/linux/kernel/v4.x/linux-4.16.3.tar.gz>



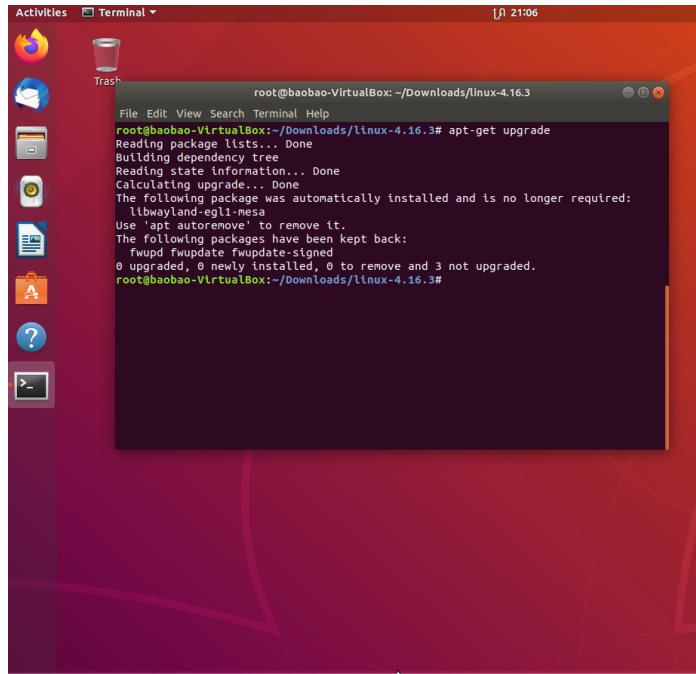
- o tar xvzf linux-4.16.3.tar.gz



- Cập nhật và nâng cấp các gói vừa tải vào hệ thống:
    - apt-get update

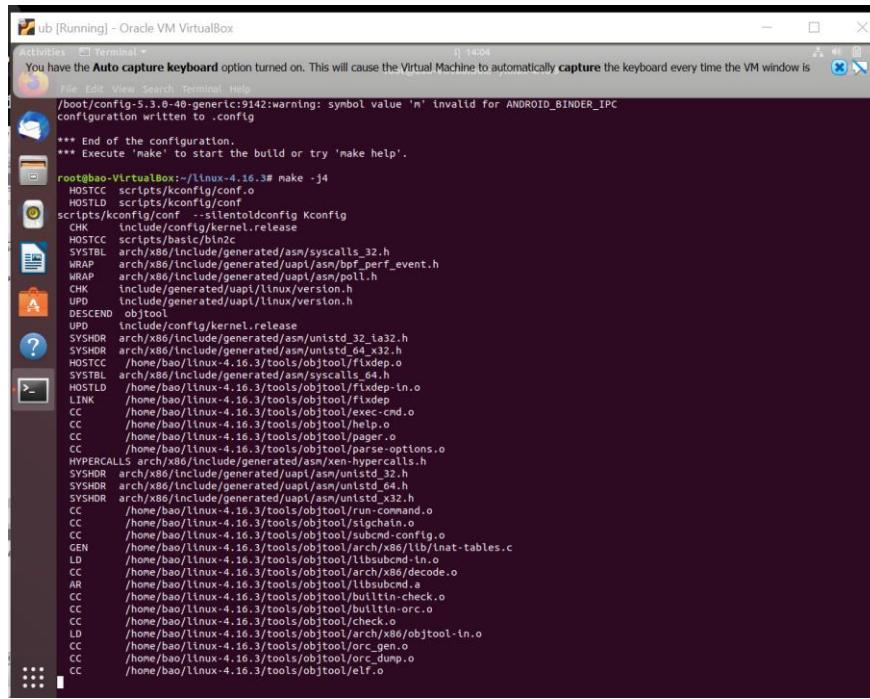


- apt-get upgrade

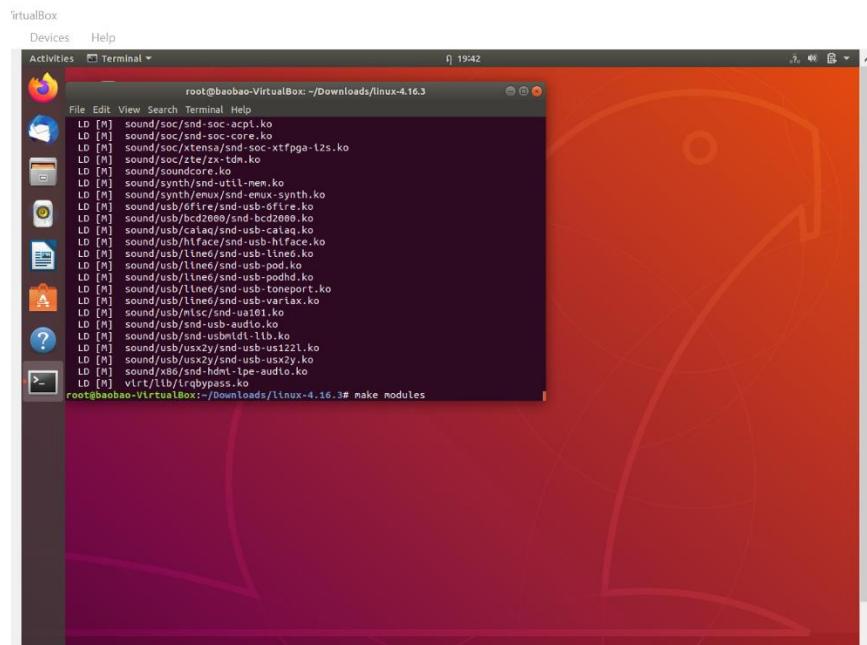


- Cài kernel vào hệ thống:

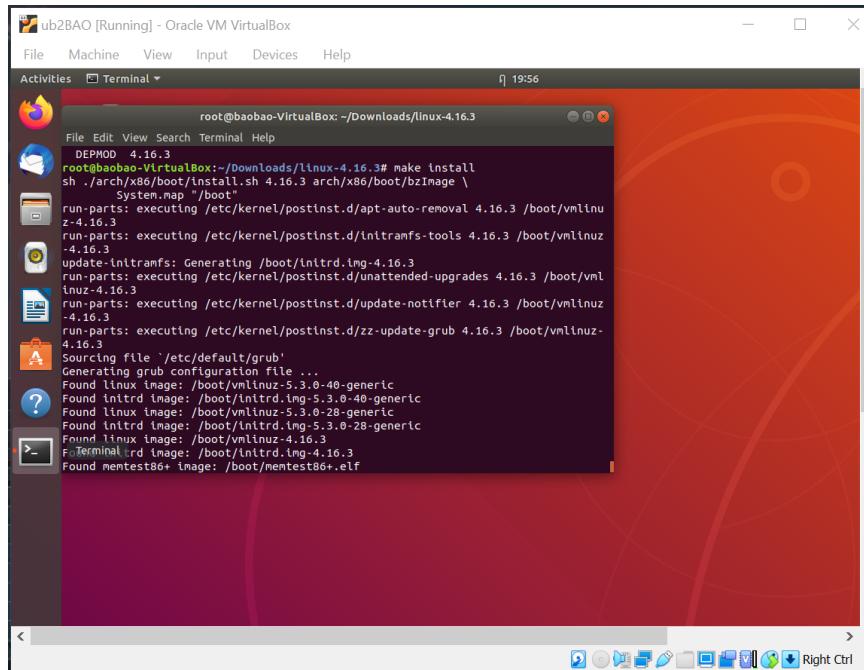
- make -j4



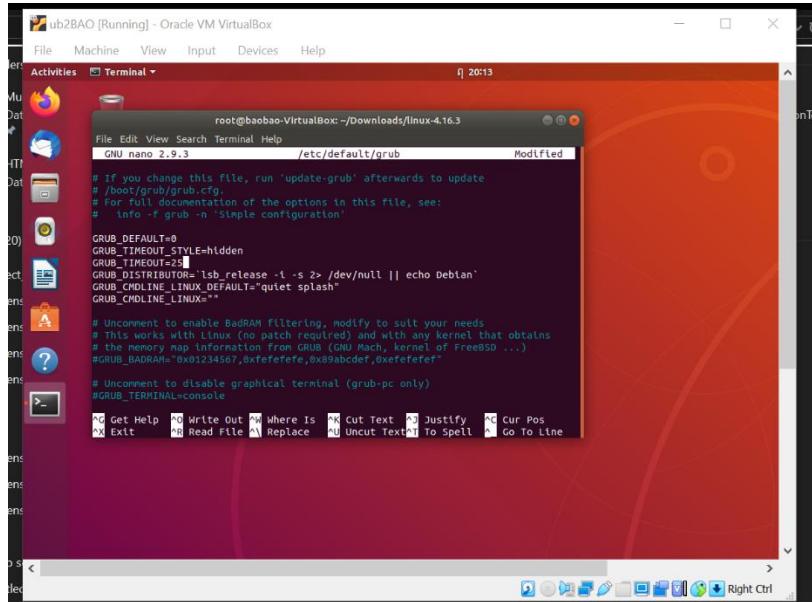
- make modules



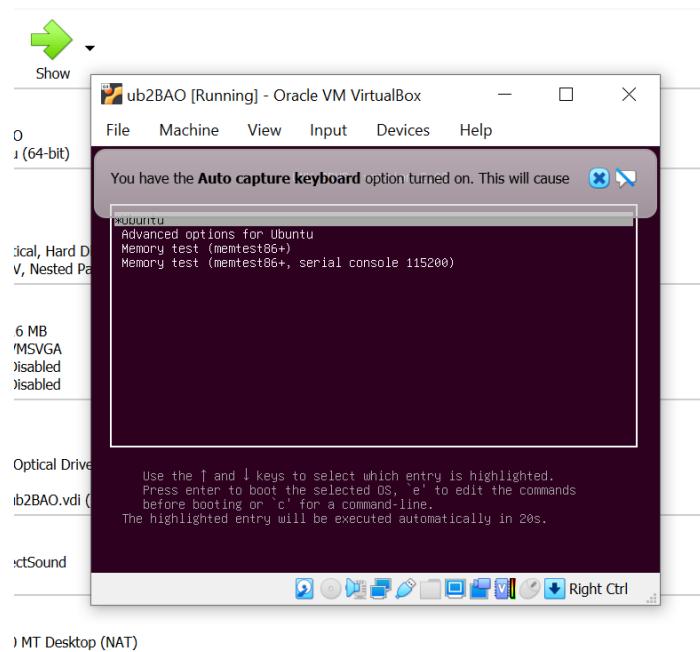
- make modules\_install
- make install



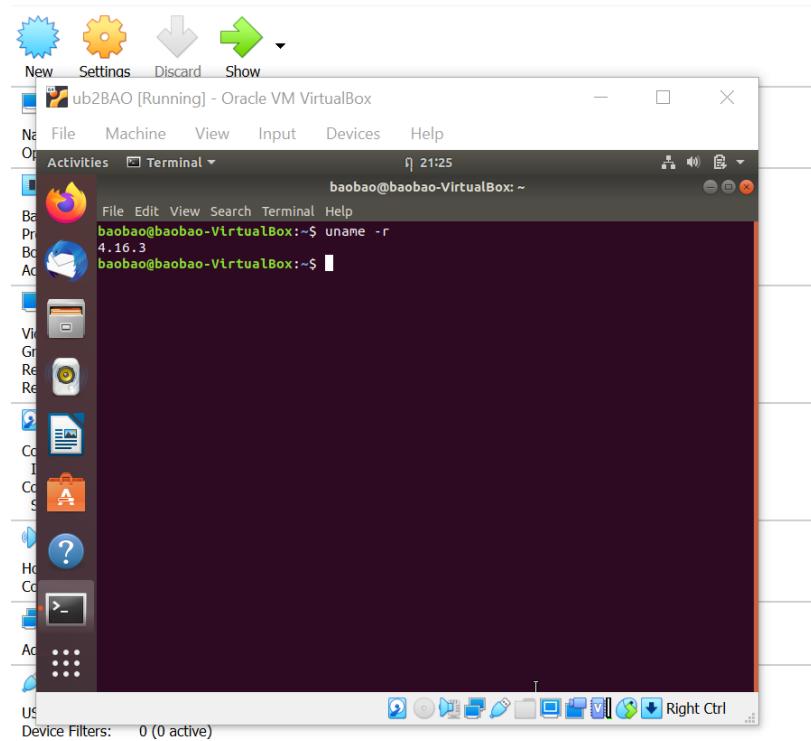
- nano /etc/default/grub (thay đổi cấu hình grub)



- reboot (reboot lại hệ thống)

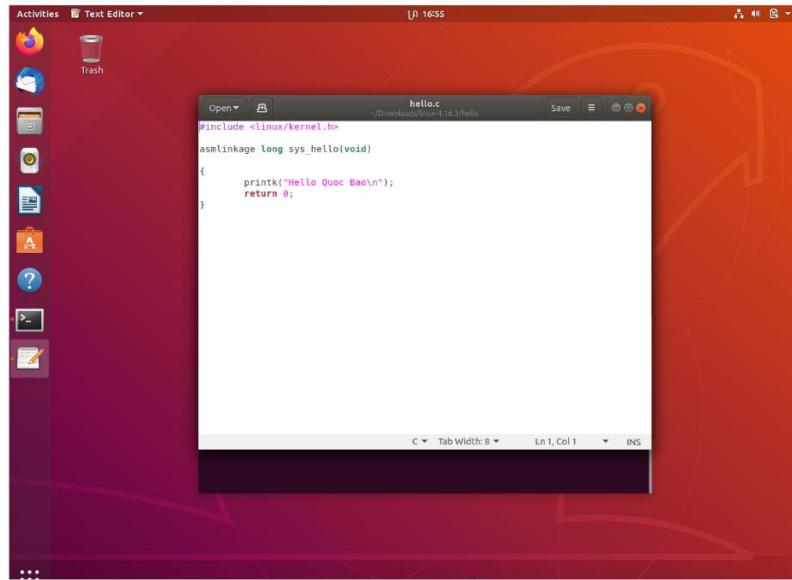


- uname -r (kiểm tra phiên bản của hệ thống)

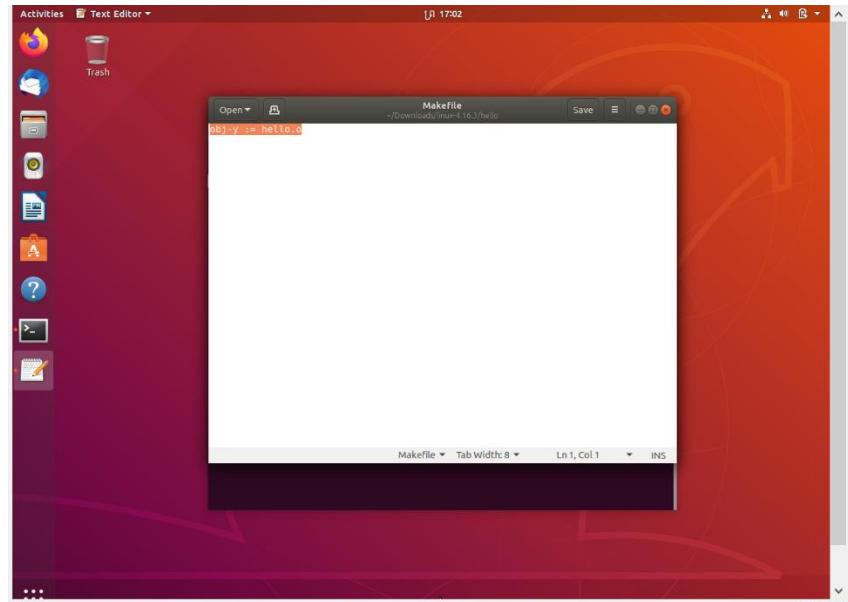


## Phần 2: Thêm lời gọi hệ thống vào Linux Kernel

- Tạo file system call tên sys\_hello()
  - o mkdir hello
  - o cd hello
  - o gedit hello.c



- gedit makefile
- Thêm “obj-y := hello.o” vào file Makefile để chắc chắn file hello.c được thêm vào kernel source code.



- Quay về thư mục cha và mở file Makefile
  - o cd ..
  - o gedit Makefile
  - o core-y += kernel/ certs/ mm/ fs/ ipc/ security / crypto/ block/ hello/ (thêm “hello/” vào sau câu lệnh trong file Makefile để thông báo với trình biên dịch là system call mới nằm trong thư mục hello)

```

Makefile
[...]
export SKIP_STACK_VALIDATION

ifeq ($(KBUILD_EXTMOD),)
core-y += kernel/certs/mm/fs/ipc/security/crypto/block/hello/
endif
endif

ifeq ($(KBUILD_DEFCONFIG),)
core-y += kernel/certs/mm/fs/ipc/security/crypto/block/hello/
endif

vmlinux-objs := $(sort $(vmlinux-objs) $(patsubst %.%,%,$filter %, $(init-y) $(init-m) \
$(core-y) $(core-m) $(drivers-y) $(drivers-m) \
$(net-y) $(net-m) $(libs-y) $(libs-m) $(virt-y)))
vmlinux-allobjs := $(sort $(vmlinux-objs) $(patsubst %.%,%,$filter %, \
$(init-) $(core-) $(drivers-) $(net-) $(libs-) $(virt-)))

init-y := $(patsubst %.%,%,$filter %, $(init-y))
core-y := $(patsubst %.%,%,$filter %, $(core-y))
drivers-y := $(patsubst %.%,%,$filter %, $(drivers-y))
net-y := $(patsubst %.%,%,$filter %, $(net-y))
libs-y := $(patsubst %.%,%,$filter %, $(libs-y))
virt-y := $(patsubst %.%,%,$filter %, $(virt-y))

# Externally visible symbols (used by link-vmlinux.sh)
export KBUILD_VMLINUX_INIT := $(head-y) $(init-y)
export KBUILD_VMLINUX_MAIN := $(core-y) $(libs-y) $(drivers-y) $(net-y) $(virt-y)
export KBUILD_VMLINUX_LIBS := $(libs-y)

[...]

```

- Vào thư mục syscalls, mở file syscall\_64.tbl:
  - o cd arch/x86/entry/syscalls/
  - o gedit syscall\_64.tbl
  - o thêm dòng “548 64 hello sys\_hello” vào cuối

```

root@baobao-VirtualBox:~/Downloads/linux-4.16.3# cd arch/x86/entry/syscalls
root@baobao-VirtualBox:~/Downloads/linux-4.16.3/arch/x86/entry/syscalls# gedit syscall_64.tbl
[...]
521 x32 ptrace compat.sys_ptrace
522 x32 rt_sigpending compat.sys_rt_sigpending
523 x32 rt_sigtimedwait compat.sys_rt_sigtimedwait
524 x32 rt_sigsuspend compat.sys_rt_sigsuspend
525 x32 signalstack compat.sys.signalstack
526 x32 timer_create compat.sys.timer_create
527 x32 mq_notify compat.sys.mq_notify
528 x32 kexec_load compat.sys.Kexec_load
529 x32 sched_setid compat.sys.sched_setid
530 x32 set_robust_list compat.sys.set_robust_list
531 x32 get_robust_list compat.sys.get_robust_list
532 x32 vmsplice compat.sys.vmsplice
533 x32 move_pages compat.sys.move_pages
534 x32 preadv compat.sys.preadv
535 x32 pwritev compat.sys.pwritev
536 x32 rt_tgsisqueueinfo compat.sys.rt_tgsisqueueinfo
537 x32 recvmmsg compat.sys.recvmmsg
538 x32 sendmmsg compat.sys.sendmmsg
539 x32 process_vm_readv compat.sys.process_vm_readv
540 x32 process_vm_writev compat.sys.process_vm_writev
541 x32 clone compat.sys.clone
542 x32 getsockopt compat.sys.getsockopt
543 x32 io_setup compat.sys.io_setup
544 x32 io_submit compat.sys.io_submit
545 x32 execveat compat.sys.execveat_ptregs
546 x32 preadv2 compat.sys.preadv64v2
547 x32 pwritev2 compat.sys.pwritev64v2
548 x32 hello sys_hello

[...]

```

- Truy cập vào thư mục linux và mở file syscalls.h:
  - o cd include/linux/
  - o gedit syscalls.h
  - o Thêm dòng “asmlinkage long sys\_hello(void)” vào cuối tệp trên lệnh “endif”

```

root@baobao-VirtualBox:~/Downloads/linux-4.16.3/include/linux
File Edit View Search Terminal Help
root@baobao-VirtualBox:~/Downloads/linux-4.16.3# cd include/linux
root@baobao-VirtualBox:~/Downloads/linux-4.16.3/include/linux# gedit syscalls.h
[...]
unsigned long sys_finit_module(int fd, const char *uargs, int flags);
asm linkage long sys_seccomp(unsigned int op, unsigned int flags,
                           const char *path, size_t count);
asm linkage long sys_getrandom(char *buf, size_t count,
                           unsigned int flags);
asm linkage long sys_bpf(int cmd, union bpf_attr *attr, unsigned int size);
asm linkage long sys_execveat(int fd, const char *filename,
                           const char *const *argp,
                           const char *const *envp, int flags);

asm linkage long sys_mbarrier(int cmd, int flags);
asm linkage long sys_copy_file_range(int fd_in, loff_t __user *off_in,
                                    int fd_out, loff_t __user *off_out,
                                    size_t len, unsigned int flags);
asm linkage long sys_clock2(unsigned long start, size_t len, int flags);
asm linkage long sys_pkey_mprotect(unsigned long start, size_t len,
                                 unsigned long prot, int pkey);
asm linkage long sys_pkey_alloc(unsigned long flags, unsigned long init_val);
asm linkage long sys_pkey_free(int pkey);
asm linkage long sys_statx(int fd, const char __user *path, unsigned flags,
                         unsigned mask, struct statx __user *buffer);
asm linkage long sys_hello(void);

#endif

```

C/ObjC Header ▾ Tab Width: 8 ▾ Ln 943, Col 1 ▾ INS

- Quay về home và tạo file userspace.c:
  - o cd ~ (quay về home)
  - o gedit userspace.c (tạo file userspace.c)

```

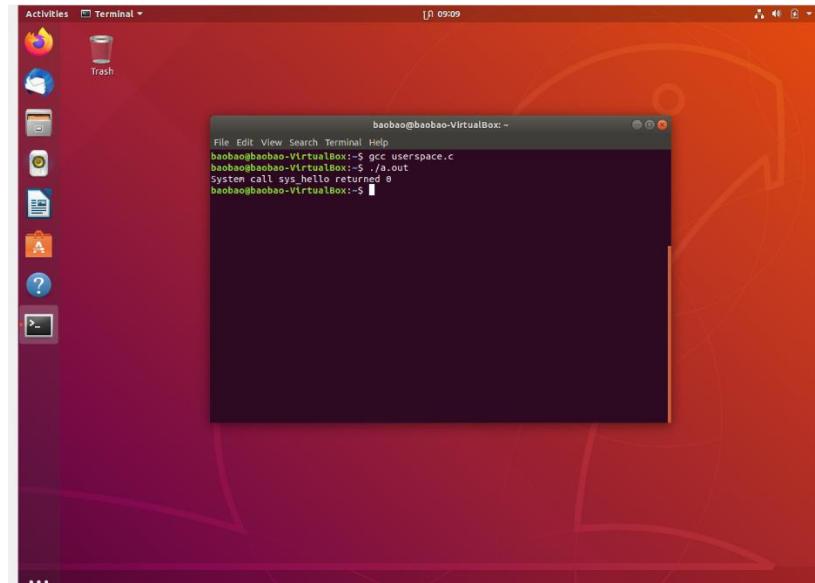
Activities Text Editor ▾
Open ▾ userspace.c
Save
[...] 06:17
#include <stdio.h>
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <unistd.h>

int main()
{
    long int amma = syscall(548);
    printf("System call sys_hello returned %ld\n", amma);
    return 0;
}

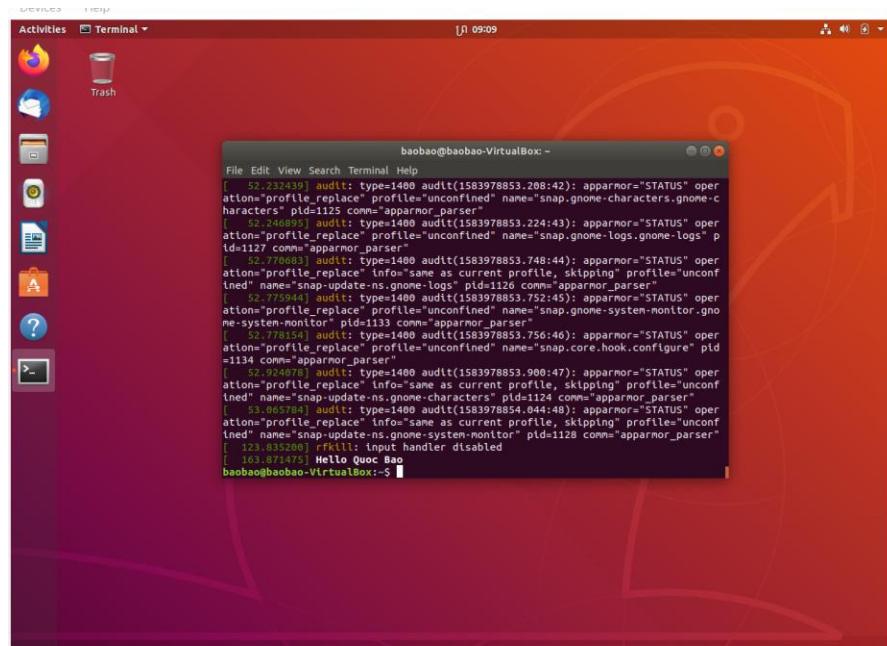
```

C ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

- Chạy file userspace.c:
  - o gcc userspace.c
  - o ./a.out
  - o Nếu xuất hiện dòng “System call sys\_hello returned 0” là đúng
  - o Ngược lại, xuất hiện dòng “System call sys\_hello returned -1” là sai



- dmesg (in lời nhắn ra terminal)



### Phần 3: Làm Việc Với Tiến Trình

- Những câu lệnh của phần 3 sẽ tương tự và cùng mục đích như phần 2 , ngoại trừ một số khác biệt.
- Tạo file system\_call tên sys\_print\_self():
  - o mkdir printself
  - o cd printself

- gedit printself.c
- câu lệnh “task->comm”: để lấy tên chương trình
- câu lệnh “task->pid”: để lấy process id
- câu lệnh “task->state”: để lấy trạng thái tiến trình

The screenshot shows a Linux desktop environment with a Unity interface. On the left, there's a dock with icons for various applications like a browser, file manager, and terminal. A terminal window is open at the top, showing root access to a VirtualBox machine. The command entered is:

```
root@baobao-VirtualBox:~/Downloads/linux-4.16.3/printself# mkdir printself
root@baobao-VirtualBox:~/Downloads/linux-4.16.3# cd printself
root@baobao-VirtualBox:~/Downloads/linux-4.16.3/printself# gedit printself.c
```

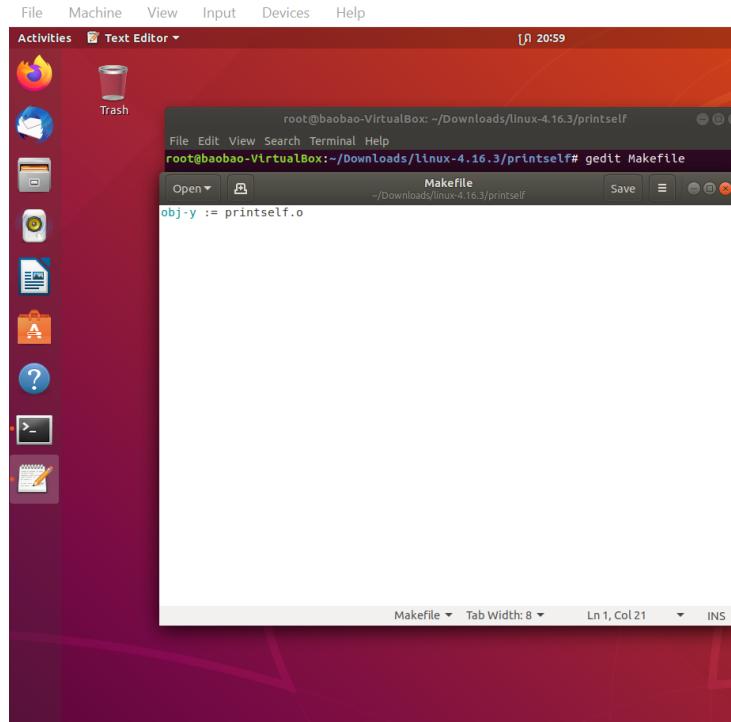
Below the terminal is a text editor window titled "printself.c". It contains the following C code:

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

asm linkage long sys_print_self(void)
{
    struct task_struct *task;
    for (task = current; task != &init_task; task = task->parent) {
        printk("PROCESS %s\n", task->comm);
        printk("PID NUMBER: %ld\n", (long) task->pid);
        printk("Process State: %ld\n", (long) task->state);
    }
    return 0;
}
```

The text editor has standard features like "Open", "Save", and status bars showing "Tab Width: 8", "Ln 13, Col 2", and "INS".

- gedit Makefile rồi thêm “obj-y := printself.o”



- Quay về thư mục cha linux-4.16.3 rồi mở file Makefile:
  - o cd ..
  - o gedit Makefile
  - o Tìm kiếm “core-y += kernel/ certs/ mm/ fs/ ipc/ security / crypto/ block/ **printself/**” (thêm printself vào)

```

root@baobao-VirtualBox:~/Downloads/linux-4.16.3#
File Edit View Search Terminal Help
root@baobao-VirtualBox:~/Downloads/linux-4.16.3# gedit Makefile
Open Makefile ~/Downloads/linux-4.16.3 Save
endif
endif

ifeq ($(KBUILD_EXTMOD),
core-y      += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/
printself/

vmlinux-dirs := $(patsubst %,%,$(filter %, $(init-y) $(init-m) \
$(core-y) $(core-m) $(drivers-y) $(drivers-m) \
$(net-y) $(net-m) $(libs-y) $(libs-m) $(virt-y)))
vmlinux-alldirs := $(sort $(vmlinux-dirs) $(patsubst %,%,$(filter %, \
$(init-) $(core-) $(drivers-) $(net-) $(libs-) $(virt-)))

init-y      := $(patsubst %, %built-in.o, $(init-y))
core-y      := $(patsubst %, %built-in.o, $(core-y))
drivers-y   := $(patsubst %, %built-in.o, $(drivers-y))
net-y       := $(patsubst %, %built-in.o, $(net-y))
libs-y1     := $(patsubst %, %lib.a, $(libs-y))
libs-y2     := $(filter-out %a, $(patsubst %, %built-in.o, $(libs-y)))
virt-y      := $(patsubst %, %built-in.o, $(virt-y))

# Externally visible symbols (used by link-vmlinux.sh)
export KBUILD_VMLINUX_INIT := $(head-y) $(init-y)
export KBUILD_VMLINUX_MAIN := $($core-y) $(libs-y2) $(drivers-y) $(net-y) $(virt-y)

```

Makefile Tab Width: 8 Ln 985, Col 83 INS

- Vào thư mục syscalls, mở file syscall\_64.tbl:
  - o cd arch/x86/entry/syscalls/
  - o gedit syscall\_64.tbl
  - o thêm dòng “548 64 hello sys\_print\_self” vào cuối

521	x32	ptrace	compat_sys_ptrace	
522	x32	rt.sigpending	compat_sys_rt_sigpending	
523	x32	rt.sigtimedwait	compat_sys_rt_sigtimedwait	
524	x32	rt.sigqueueinfo	compat_sys_rt_sigqueueinfo	
525	x32	sigaltstack	compat_sys_sigaltstack	
526	x32	timer_create	compat_sys_timer_create	
527	x32	mq_notify	compat_sys_mq_notify	
528	x32	kexec_load	compat_sys_kexec_load	
529	x32	exit_group	compat_sys_exit_group	
530	x32	set_robust_list	compat_sys_set_robust_list	
531	x32	get_robust_list	compat_sys_get_robust_list	
532	x32	vmsplice	compat_sys_vmsplice	
533	x32	move_pages	compat_sys_move_pages	
534	x32	preadv	compat_sys_preadv64	
535	x32	pwrite	compat_sys_pwritev64	
536	x32	rt.tgsigqueueinfo	compat_sys_rt_tgsigqueueinfo	
537	x32	recvmmsg	compat_sys_recvmmsg	
538	x32	sendmmsg	compat_sys_sendmmsg	
539	x32	process_vm_readv	compat_sys_process_vm_readv	
540	x32	process_vm_writev	compat_sys_process_vm_writev	
541	x32	setssockopt	compat_sys_setssockopt	
542	x32	getsockopt	compat_sys_getsockopt	
543	x32	io_setup	compat_sys_io_setup	
544	x32	io_submit	compat_sys_io_submit	
545	x32	execveat	compat_sys_execveat/ptregs	
546	x32	preadv2	compat_sys_preadv64v2	
547	x32	pwritev2	compat_sys_pwritev64v2	
548	64	printself	sys_print_self	

Plain Text Tab Width: 8 Ln 383, Col 1 INS

- Truy cập vào thư mục linux và mở file syscalls.h:
  - o cd include/linux/
  - o gedit syscalls.h
  - o Thêm dòng “asmlinkage long sys\_print\_self(void)” vào cuối tệp trên lệnh “endif”

```

Activities   Text Editor
Open   Save
sy... /Downloads/linux-4.18.3/include/linux
Save
sy... 22:27
unsigned long idx1, unsigned long idx2);
asmlinkage long sys_finit_module(int fd, const char __user *args, int flags);
asmlinkage long sys_seccomp(int op, unsigned int flags,
                           const char __user *args);
asmlinkage long sys_getrandom(char __user *buf, size_t count,
                             unsigned int flags);
asmlinkage long sys_bpf(int cmd, union bpf_attr *attr, unsigned int size);
asmlinkage long sys_execveat(int dfd, const char __user *filename,
                           const char __user *const __user *argv,
                           const char __user *const __user *envp, int flags);
asmlinkage long sys_mbarrier(int cmd, int flags);
asmlinkage long sys_copy_file_range(int fd_in, loff_t __user *off_in,
                                   int fd_out, loff_t __user *off_out,
                                   size_t len, unsigned int flags);
asmlinkage long sys_mlock2(unsigned long start, size_t len, int flags);
asmlinkage long sys_pkey_mprotect(unsigned long start, size_t len,
                                 unsigned long prot, int pkey);
asmlinkage long sys_pkey_alloc(unsigned long flags, unsigned long init_val);
asmlinkage long sys_pkey_free(int pkey);
asmlinkage long sys_statx(int dfd, const char __user *path, unsigned flags,
                         unsigned mask, struct statx __user *buffer);
asmlinkage long sys_print_self(void);

#endif

```

- Vì đây là tạo một syscall mới nên ta phải chạy cập nhật, biên dịch lại kernel:
  - o apt-get update
  - o apt-get upgrade
  - o make -j4
  - o make modules\_install install
  - o shutdown -r now (reboot lại máy)
- Quay về home và tạo thư mục userspace.c:
  - o cd ~
  - o gedit userspace

A screenshot of a Linux desktop environment (Ubuntu) showing a terminal window with root privileges. The terminal window title is "root@baobao-VirtualBox:~". The command entered is "gedit userspace.c". The code in the file is:

```
#include <studio.h>
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <unistd.h>

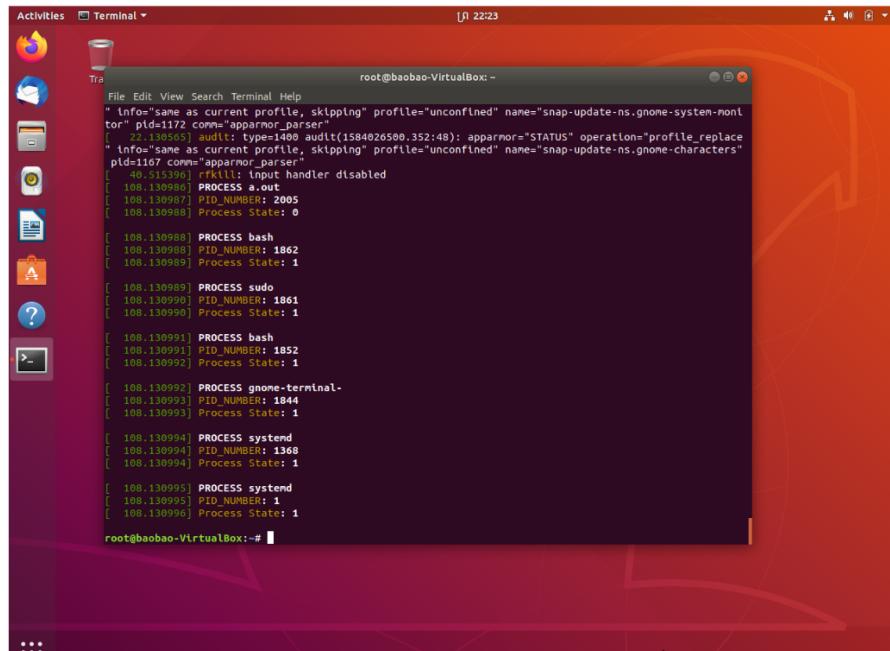
int main()
{
    long int amma = syscall(540);
    printf("System call sys_print_self returned %ld \n", amma);
    return 0;
}
```

The terminal shows the output of the program: "System call sys\_print\_self returned 0".

- Chạy file userspace.c:
  - o gcc userspace.c
  - o ./a.out

A screenshot of a Linux desktop environment (Ubuntu) showing a terminal window with root privileges. The terminal window title is "root@baobao-VirtualBox:~". The command entered is "gcc userspace.c" followed by "./a.out". The output of the program is displayed in the terminal.

- dmesg (in lời nhắn ra màn hình)



A screenshot of an Ubuntu desktop environment. In the center is a terminal window titled "root@baobao-VirtualBox: ~" with the status bar showing "J 22:23". The terminal displays a list of processes from the "top" command:

```
File Edit View Search Terminal Help
root@baobao-VirtualBox: ~
"Info='same as current profile, skipping' profile='unconfined' name='snap-update-ns.gnome-system-monitor' pid=172 comm='apparmor_parser'
[ 40.515396] rtkill: input handler disabled
[ 108.130986] PROCESS a.out
[ 108.130987] PID_NUMBER: 2005
[ 108.130988] Process State: 0
[ 108.130988] PROCESS bash
[ 108.130988] PID_NUMBER: 1062
[ 108.130989] Process State: 1
[ 108.130989] PROCESS sudo
[ 108.130990] PID_NUMBER: 1061
[ 108.130990] Process State: 1
[ 108.130991] PROCESS bash
[ 108.130991] PID_NUMBER: 1052
[ 108.130992] Process State: 1
[ 108.130992] PROCESS gnome-terminal
[ 108.130993] PID_NUMBER: 1044
[ 108.130993] Process State: 1
[ 108.130994] PROCESS systemd
[ 108.130994] PID_NUMBER: 1068
[ 108.130994] Process State: 1
[ 108.130995] PROCESS systemd
[ 108.130995] PID_NUMBER: 1
[ 108.130996] Process State: 1
root@baobao-VirtualBox: ~#
```