

# **Napredni operativni sistemi**

## **Prevođenje Linux kernela**

**Doc. dr Aleksandar Stanimirović**

Katedra za računarstvo, Elektronski fakultet u Nišu

# Oficijelna verzija kernela

- Testirano na Ubuntu 15.04 distribuciji (<https://www.ubuntu.com>)
- Oficijalna verzija kernela: 4.8.11 (<https://www.kernel.org>)
- <http://askubuntu.com/questions/163298/whats-a-simple-way-to-recompile-the-kernel>

- Instalacija neophodnih alata za prevođenje

**`sudo apt-get update`**

**`sudo apt-get install build-essential`**

- Instalacija podrške za tekstualno konfigurisanje kernela

**`sudo apt-get install libncurses5 libncurses5-dev`**

- Instalacija podrške za grafičko konfigurisanje kernela

**`sudo apt-get install libqt4-dev pkg-config`**

- Instalacija Open SSL biblioteke

**`sudo apt-get install libssl-dev`**

# Oficijelna verzija kernela

- Kreiranje direktorijum src u korisnikovom home direktorijumu home

```
cd ~
```

```
mkdir src
```

```
cd src
```

- Download koda Linux kernela

```
wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.8.11.tar.xz
```

Alternativno download iz bilo kog browser-a i kopiranje arhive u kreirani src direktorijum

- Raspakovanje arhive sa kodom

```
tar xf linux-4.8.11.tar.xz
```

- Prelazak u direktorijum sa kodom

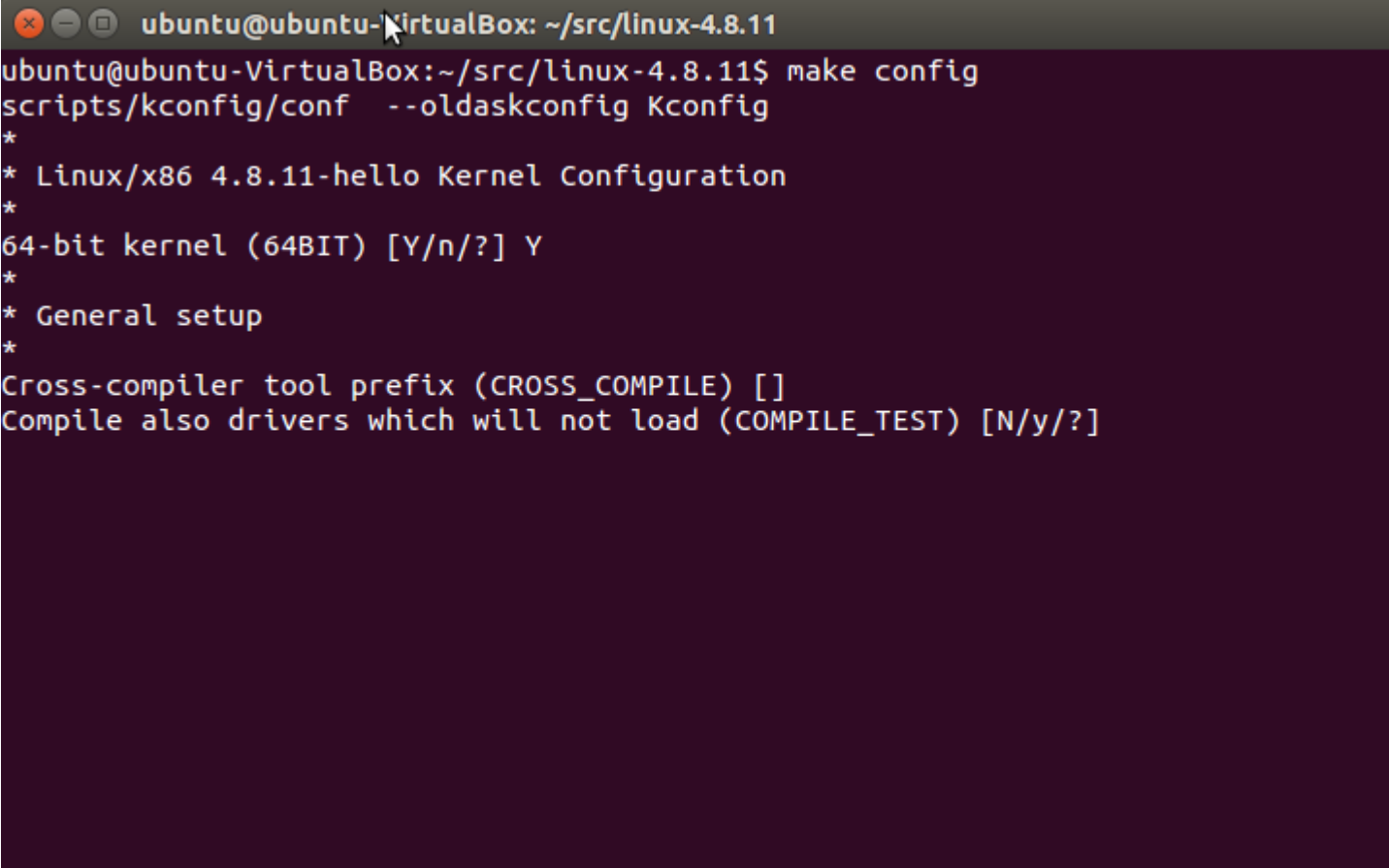
```
cd linux-4.8.11
```

# Oficijelna verzija kernela

- Pre prevođenja koda Linux kernela treba pripremiti konfiguraciju
- Konfiguracija se nalazi u direktorijumu **.config** u osnovnom direktorijumu koda Linux kernela.
- Inicijalno konfiguracija ne postoji i treba je kreirati:
  - **Prihvatanje podrazumevane konfiguracije**  
`make defconfig`
  - **Kreiranje konfiguracije korak po korak**  
`make config`
  - **Kreiranje konfiguracije u tekstualnom režimu**  
`make menuconfig`
  - **Kreiranje konfiguracije u grafičkom režimu**  
`make xconfig`

# Oficijelna verzija kernela

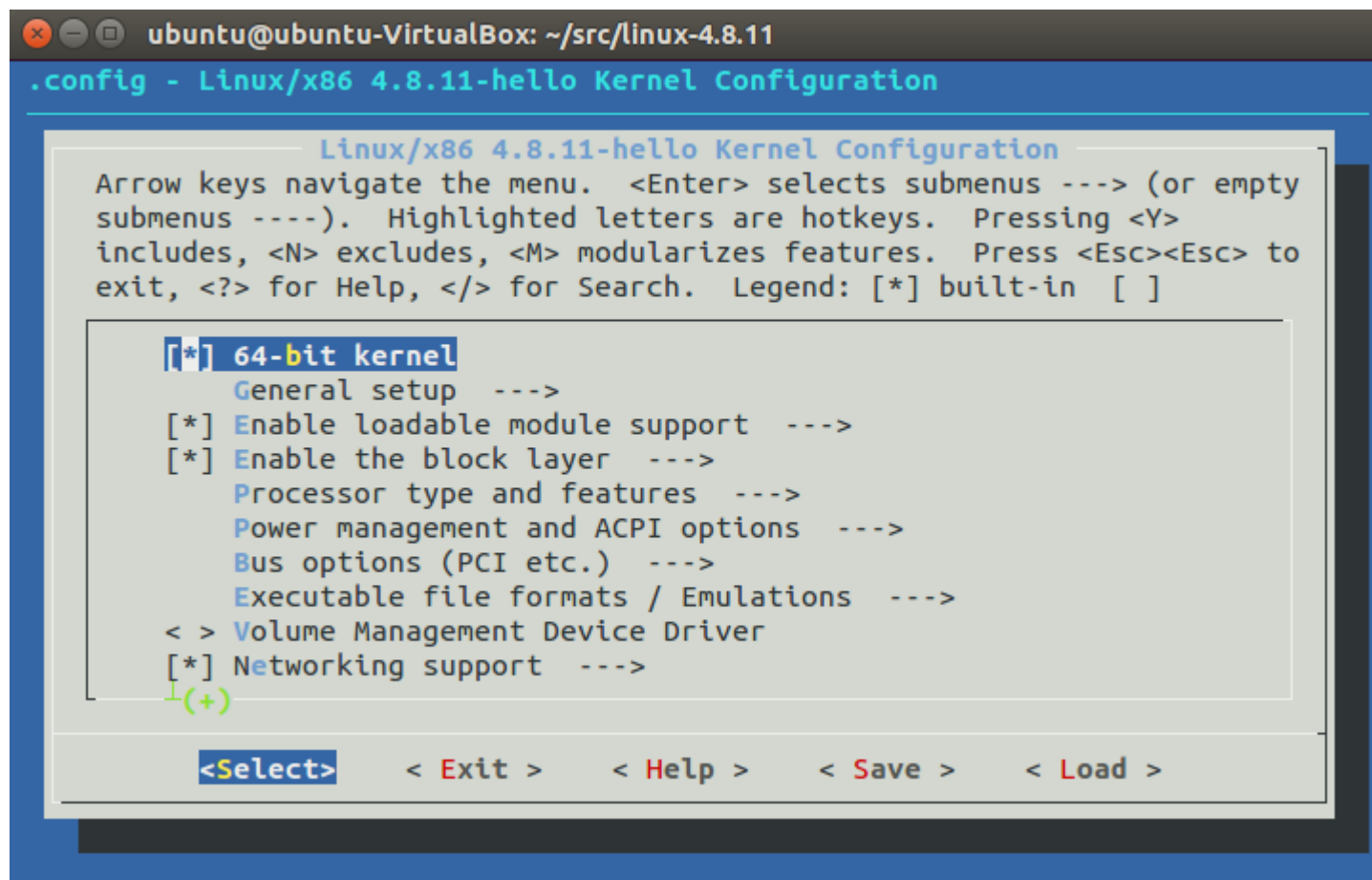
- Konfiguracija korak po korak

A terminal window with a dark purple background and light-colored text. The window title bar shows 'ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11'. The terminal content shows the execution of 'make config' and the start of the 'scripts/kconfig/conf' configuration menu. The menu displays 'Linux/x86 4.8.11-hello Kernel Configuration' and prompts for '64-bit kernel (64BIT) [Y/n/?]' (answered 'Y') and 'General setup'. It also shows 'Cross-compiler tool prefix (CROSS\_COMPILE) []' and 'Compile also drivers which will not load (COMPILE\_TEST) [N/y/?]'.

```
ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11
ubuntu@ubuntu-VirtualBox:~/src/linux-4.8.11$ make config
scripts/kconfig/conf  --oldaskconfig Kconfig
*
* Linux/x86 4.8.11-hello Kernel Configuration
*
64-bit kernel (64BIT) [Y/n/?] Y
*
* General setup
*
Cross-compiler tool prefix (CROSS_COMPILE) []
Compile also drivers which will not load (COMPILE_TEST) [N/y/?]
```

# Oficijelna verzija kernela

- Konfiguracija u tekstualnom režimu



The screenshot shows a terminal window titled 'ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11'. The main window is the 'Linux/x86 4.8.11-hello Kernel Configuration' menu. It displays instructions for navigating the menu using arrow keys, Enter, Y, N, M, Esc, and a search function. The menu items are listed with their status: [\*] 64-bit kernel (selected), General setup, [\*] Enable loadable module support, [\*] Enable the block layer, Processor type and features, Power management and ACPI options, Bus options (PCI etc.), Executable file formats / Emulations, < > Volume Management Device Driver, and [\*] Networking support. A green '(+)' symbol is visible next to the Networking support option. At the bottom, there are navigation buttons: <Select>, < Exit >, < Help >, < Save >, and < Load >.

```
ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11
.config - Linux/x86 4.8.11-hello Kernel Configuration

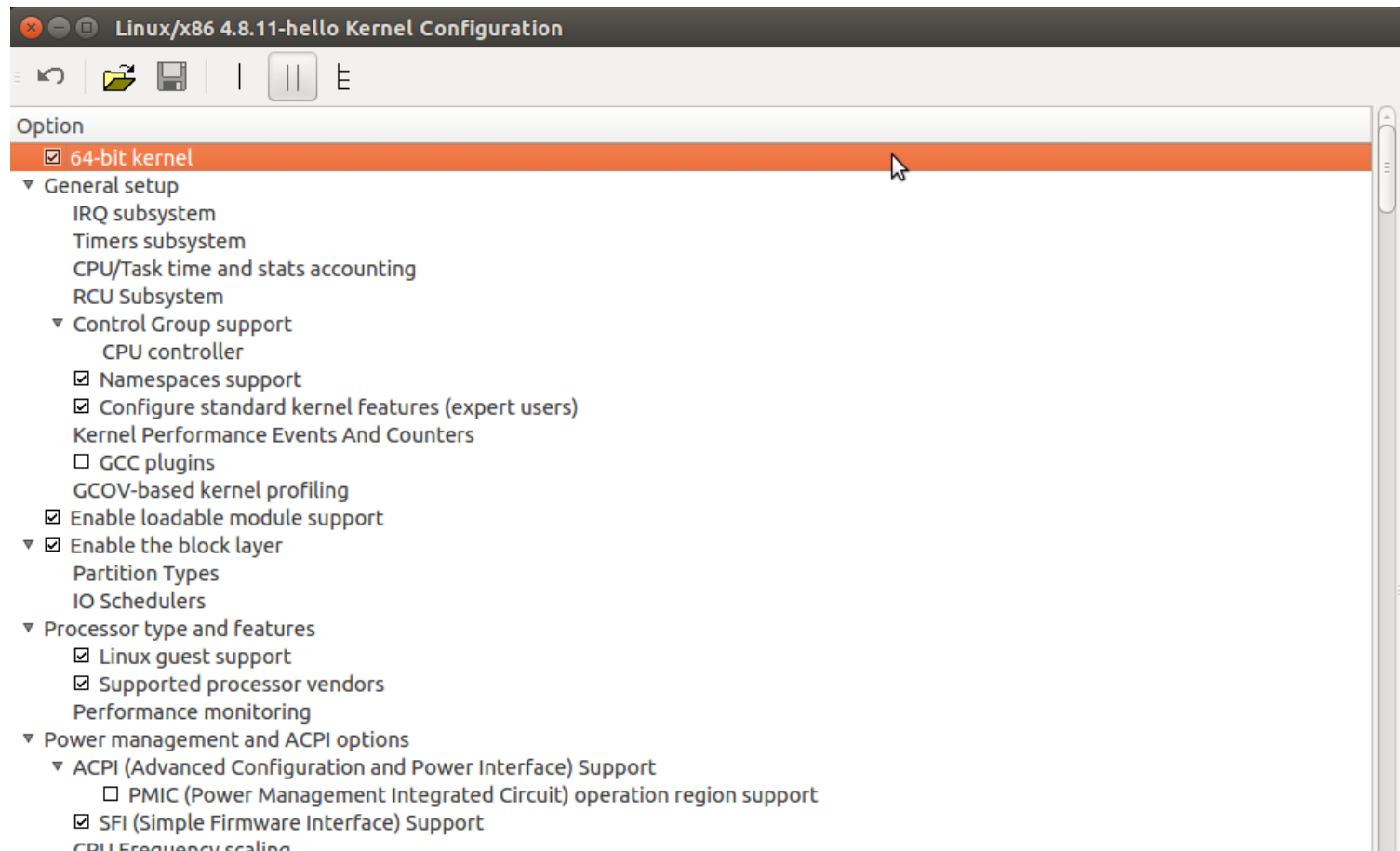
Linux/x86 4.8.11-hello Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

[*] 64-bit kernel
  General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
  Processor type and features --->
  Power management and ACPI options --->
  Bus options (PCI etc.) --->
  Executable file formats / Emulations --->
< > Volume Management Device Driver
[*] Networking support --->
  (+)

<Select>  < Exit >  < Help >  < Save >  < Load >
```

# Oficijelna verzija kernela

- Konfiguracija u grafičkom režimu



# Oficijelna verzija kernela

- Prevođenje kernela (alternativa 1)

- Prihvata se podrazumevana konfiguracija

```
make defconfig
```

- Brisanje rezultata prethodnih prevođenja (ukoliko treba krenuti od početka)

```
make clean
```

- Prevođenje kernela (na kraju prevođenja se kreiraju Debian instalacioni paketi)

```
make deb-pkg
```

- Nakon prevođenja paketi se nalaze u roditeljskom direktorijumu u odnosu na direktorijum u kome je kod Linux kernela (u konkretnom slučaju u direktorijumu src)

```
cd..
```

```
sudo dpkg -i linux-image-4.8.11*.deb
```

```
sudo dpkg -i linux-headers-4.8.11*.deb
```

- Restartovati sistem
- Prilikom podizanja sistema treba izabrati odgovarajuću verziju kernela



# Oficijelna verzija kernela

- Prevođenje kernela (alternativa 2)

- Prihvata se podrazumevana konfiguracija

```
make defconfig
```

- Brisanje rezultata prethodnih prevođenja (ukoliko treba krenuti od početka)

```
make clean
```

- Prevođenje kernela

```
make
```

- Prevođenje kernel modula

```
make modules
```

- Instalacija nove verzije kernela

```
make modules_install
```

```
make install
```

- Kreiranje RAM fajl sistema za startovanje novog kernela

```
cd /boot
```

```
sudo mkinitramfs -ko initrd.img-4.8.11 4.8.11
```

```
sudo update-grub
```

- Restartovati sistem
- Prilikom podizanja sistema treba izabrati odgovarajuću verziju kernela

# Ubuntu verzija kernela

- *Testirano na Ubuntu 15.04 distribuciji (<https://www.ubuntu.com>)*
- *Ubuntu verzija kernela: 4.8.11*
- Kreiranje direktorijum src u korisnikovom home direktorijumu home

```
cd ~
```

```
cd src
```

- **Download koda Linux kernela**

```
apt-get source linux-image-$(uname -r)
```

- **Instalacija podrške**

```
sudo apt-get build-dep linux-image-$(uname -r)
```

- **Prelazak u direktorijum sa kodom**

```
cd linux-3.19.0
```

# Ubuntu verzija kernela

- Brisanje rezultata prethodnih prevođenja (ukoliko treba krenuti od početka)

```
fakeroot debian/rules clean
```

- Prevođenje kernela (na kraju prevođenja se kreiraju Debian instalacioni paketi)

```
fakeroot debian/rules binary
```

- Nakon prevođenja paketi se nalaze u roditeljskom direktorijumu u odnosu na direktorijum u kome je kod Linux kernela (u konkretnom slučaju u direktorijumu src)

```
cd..
```

```
sudo dpkg -i linux-image-3.19.0*.deb
```

```
sudo dpkg -i linux-headers-3.19.0*.deb
```

- Restartovati sistem
- Prilikom podizanja sistema treba izabrati odgovarajuću verziju kernela

# GRUB konfiguracija

- <http://www.howtogeek.com/196655/how-to-configure-the-grub2-boot-loaders-settings/>
- GRUB – boot loader package
- Ukoliko se prilikom startovanja Linux sistema ne pojavi mogućnost izbora kernela treba izmeniti GRUB konfiguraciju.
- Konfiguracija u tekstualnoj datoteci /etc/default/grub

```
sudo nano /etc/default/grub
```

- Da bi izmene u konfiguraciji postale validne neophodno je izvršiti sledeću komandu:

```
sudo update-grub
```

# GRUB konfiguracija

- Potrebno je zakomentarisati liniju GRUB\_HIDDEN\_TIMEOUT=0 da bi se prikazala lista za izbor različitih verzija kernela
- Podešavanje GRUB\_TIMEOUT=10 određuje koliko će lista biti vidljiva pre nego što se pokrene podrazumevana verzija kernela.

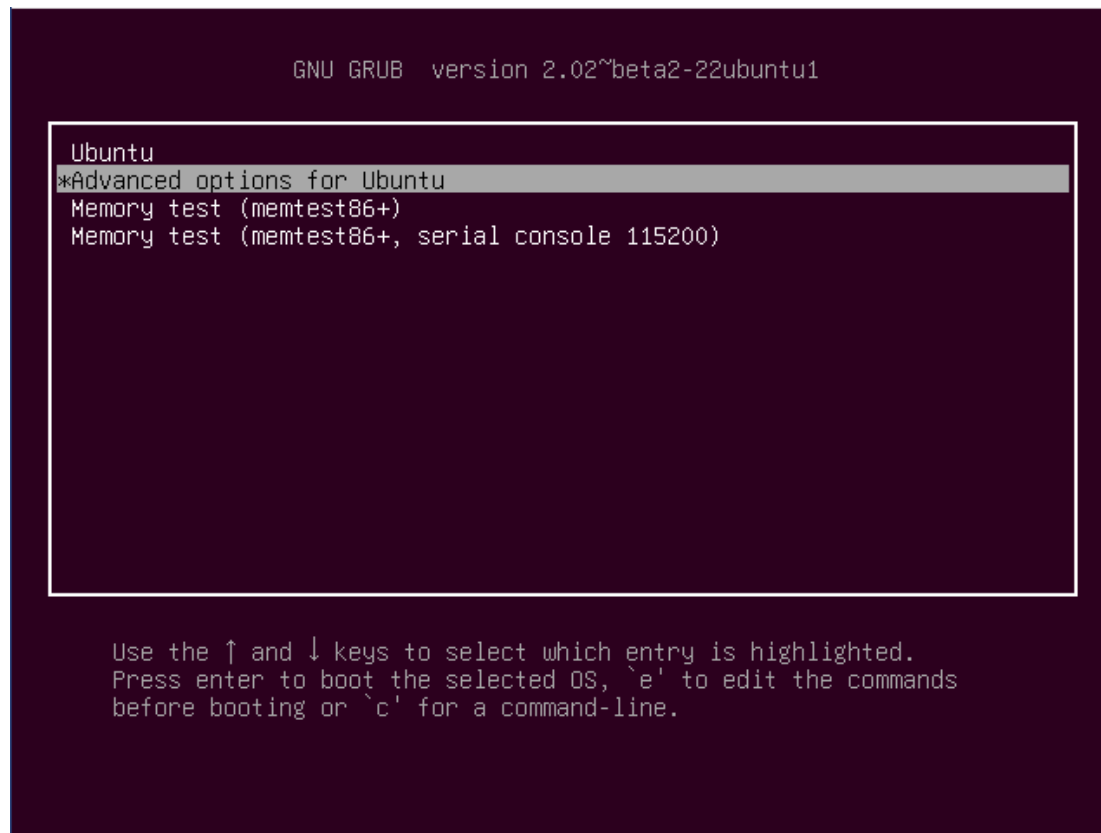
```
GRUB_DEFAULT=saved
GRUB_SAVEDefault=true

#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

GRUB_BACKGROUND="/home/chris/Pictures/grubbackground.png"
```

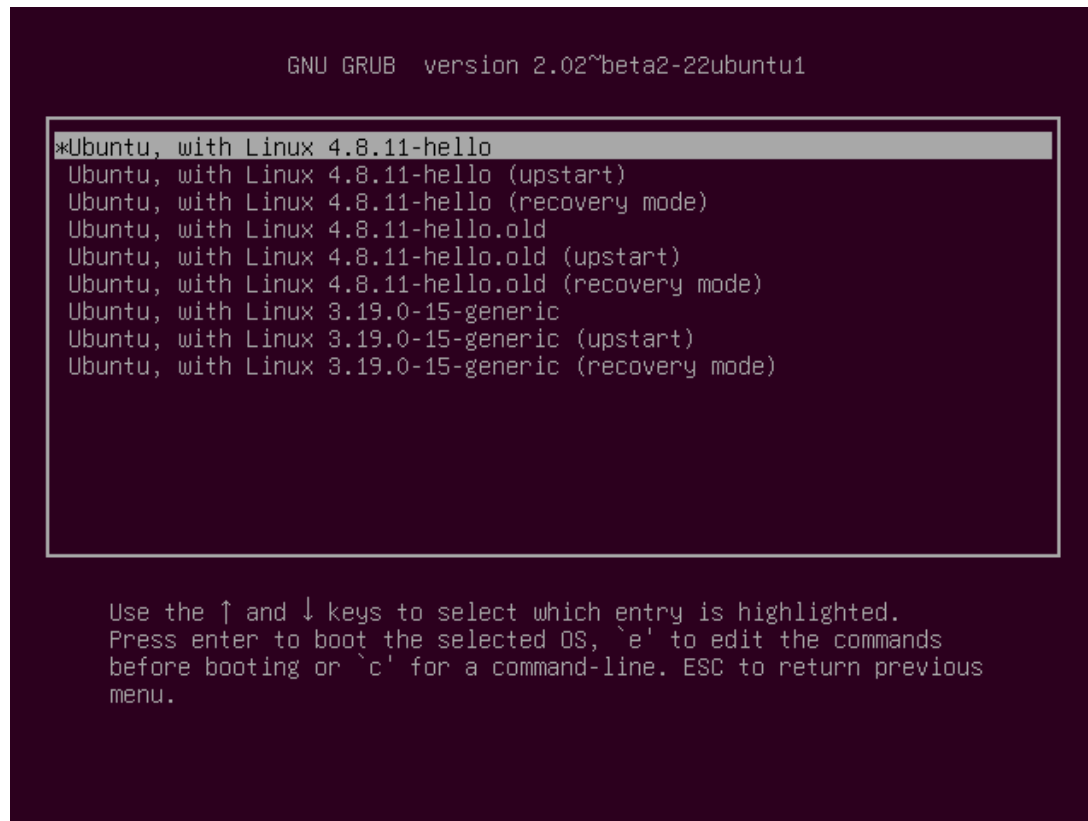
# GRUB konfiguracija

- Izbor verzije kernela prilikom startovanja operativnog sistema



# GRUB konfiguracija

- Izbor verzije kernela prilikom startovanja operativnog sistema



```
GNU GRUB version 2.02~beta2-22ubuntu1

*Ubuntu, with Linux 4.8.11-hello
  Ubuntu, with Linux 4.8.11-hello (upstart)
  Ubuntu, with Linux 4.8.11-hello (recovery mode)
  Ubuntu, with Linux 4.8.11-hello.old
  Ubuntu, with Linux 4.8.11-hello.old (upstart)
  Ubuntu, with Linux 4.8.11-hello.old (recovery mode)
  Ubuntu, with Linux 3.19.0-15-generic
  Ubuntu, with Linux 3.19.0-15-generic (upstart)
  Ubuntu, with Linux 3.19.0-15-generic (recovery mode)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e` to edit the commands
before booting or `c` for a command-line. ESC to return previous
menu.
```

# Dodavanje novog sistemskog poziva

- *Testirano na Ubuntu 15.04 distribuciji* (<https://www.ubuntu.com>)
- *Oficijalna verzija kernela: 4.8.11* (<https://www.kernel.org>)
- <https://tssurya.wordpress.com/2014/08/19/adding-a-hello-world-system-call-to-linux-kernel-3-16-0/>
- <http://web2.clarkson.edu/class/cs444/cs444.sp2015/labs/lab04.pdf>
- <http://www.franksthinktank.com/howto/addsyscall/>



# Dodavanje novog sistemskog poziva

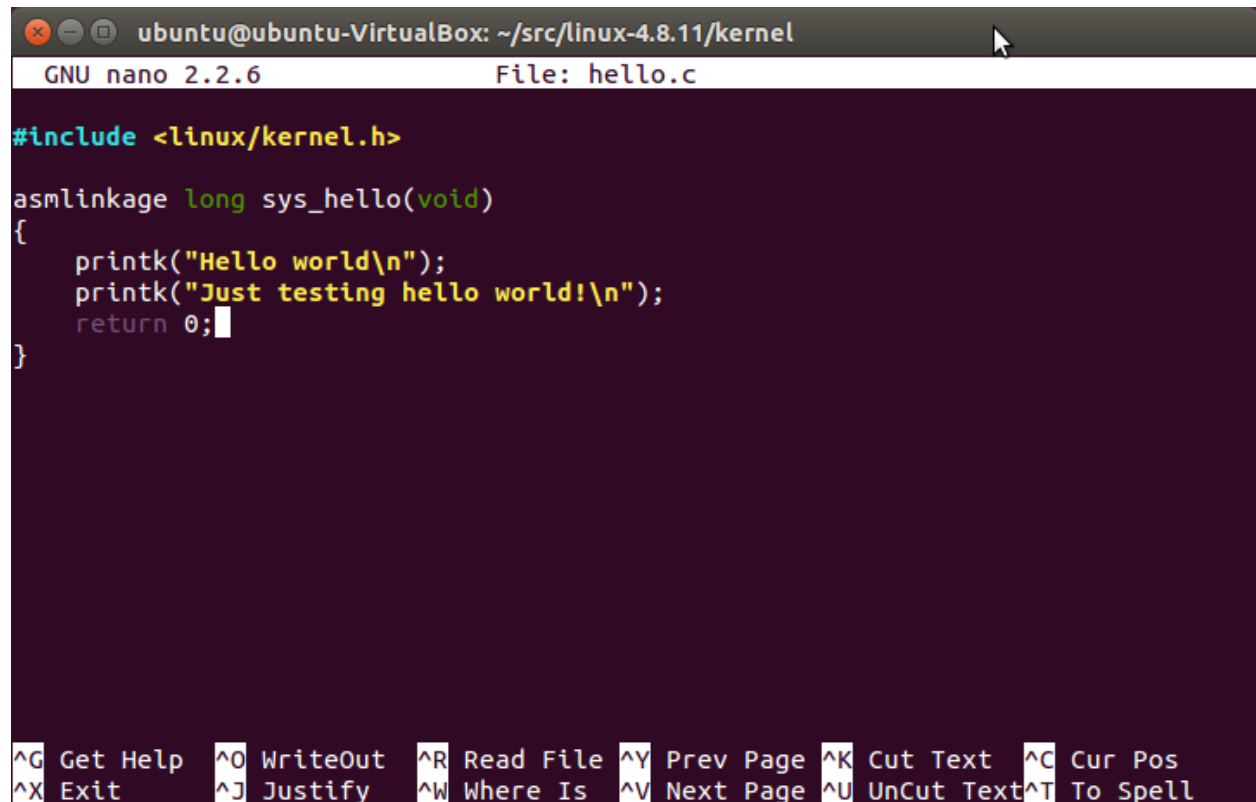
- Dodavanje novog sistemskog poziva

```
cd ~
```

```
mkdir src
```

```
cd src/linux-4.8.11/kernel
```

```
nano hello.c
```



```
ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11/kernel
GNU nano 2.2.6 File: hello.c

#include <linux/kernel.h>

asmlinkage long sys_hello(void)
{
    printk("Hello world\n");
    printk("Just testing hello world!\n");
    return 0;
}

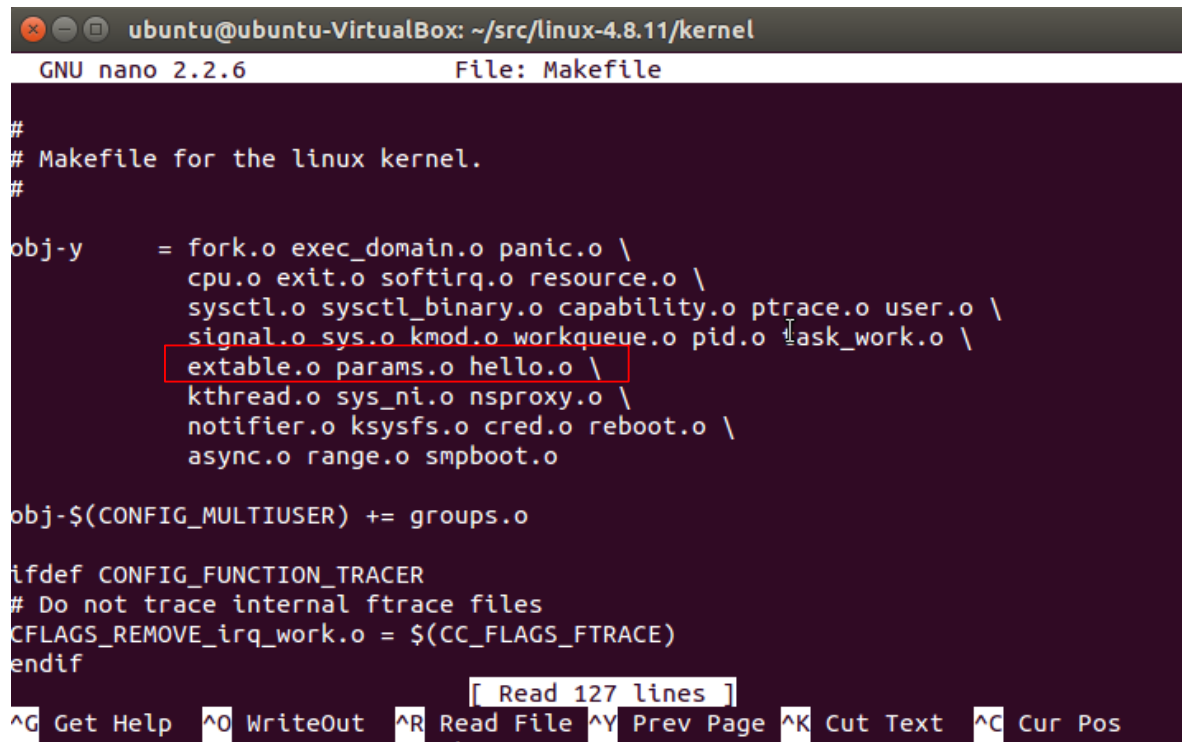
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

# Dodavanje novog sistemskog poziva

- Izmena make datoteke (~/.src/linux-4.8.11/kernel/Makefile)

nano Makefile

- Dodati hello.o datoteku u obj-y listu



```
ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11/kernel
GNU nano 2.2.6 File: Makefile

#
# Makefile for the linux kernel.
#

obj-y      = fork.o exec_domain.o panic.o \
             cpu.o exit.o softirq.o resource.o \
             sysctl.o sysctl_binary.o capability.o ptrace.o user.o \
             signal.o sys.o kmod.o workqueue.o pid.o task_work.o \
             extable.o params.o hello.o \
             kthread.o sys_ni.o nsproxy.o \
             notifier.o ksysfs.o cred.o reboot.o \
             async.o range.o smptboot.o

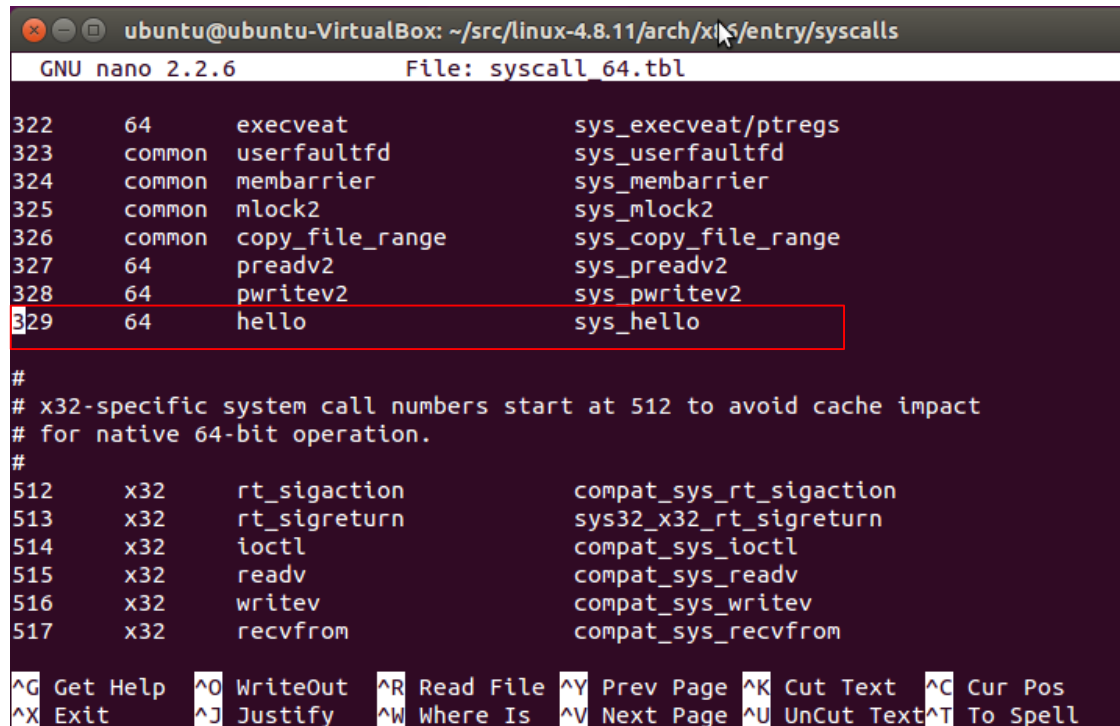
obj-$(CONFIG_MULTIUSER) += groups.o

ifdef CONFIG_FUNCTION_TRACER
# Do not trace internal ftrace files
CFLAGS_REMOVE_irq_work.o = $(CC_FLAGS_FTRACE)
endif

[ Read 127 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
```

# Dodavanje novog sistemskog poziva

- Izmena tabele sistemskih poziva
  - x64 operativni sistem `~/src/linux-4.8.11/arch/x86/entry/syscalls/syscall_64.tbl`
  - Novi sistemski poziv se dodaje na kraj liste sistemskih poziva. Uzima se prvi sledeći slobodan broj za novi sistemski poziv



```
ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11/arch/x86/entry/syscalls
GNU nano 2.2.6 File: syscall_64.tbl

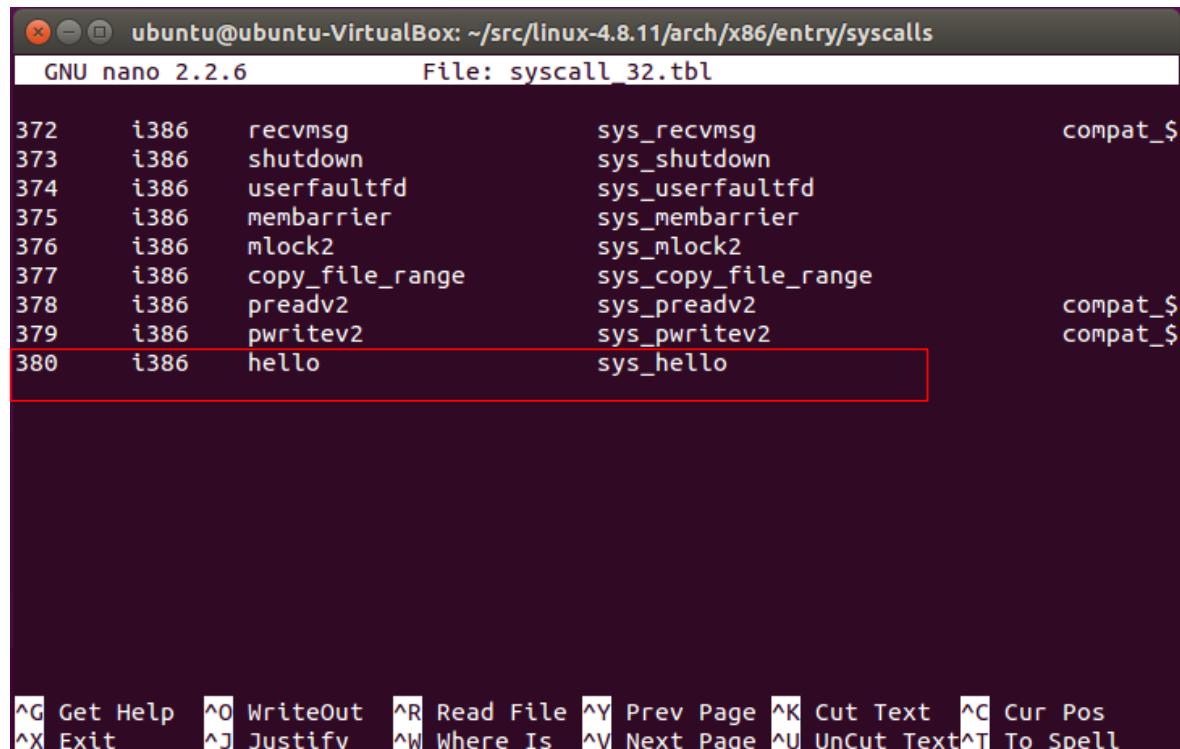
322      64      execveat      sys_execveat/ptregs
323      common userfaultfd  sys_userfaultfd
324      common membarrier  sys_membarrier
325      common mlock2      sys_mlock2
326      common copy_file_range sys_copy_file_range
327      64      preadv2      sys_preadv2
328      64      pwritev2     sys_pwritev2
329      64      hello        sys_hello

#
# x32-specific system call numbers start at 512 to avoid cache impact
# for native 64-bit operation.
#
512      x32      rt_sigaction  compat_sys_rt_sigaction
513      x32      rt_sigreturn  sys32_x32_rt_sigreturn
514      x32      ioctl        compat_sys_ioctl
515      x32      readv        compat_sys_readv
516      x32      writev       compat_sys_writev
517      x32      recvfrom     compat_sys_recvfrom

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

# Dodavanje novog sistemskog poziva

- Izmena tabele sistemskih poziva
  - x86 operativni sistem `~/src/linux-4.8.11/arch/x86/entry/syscalls/syscall_32.tbl`



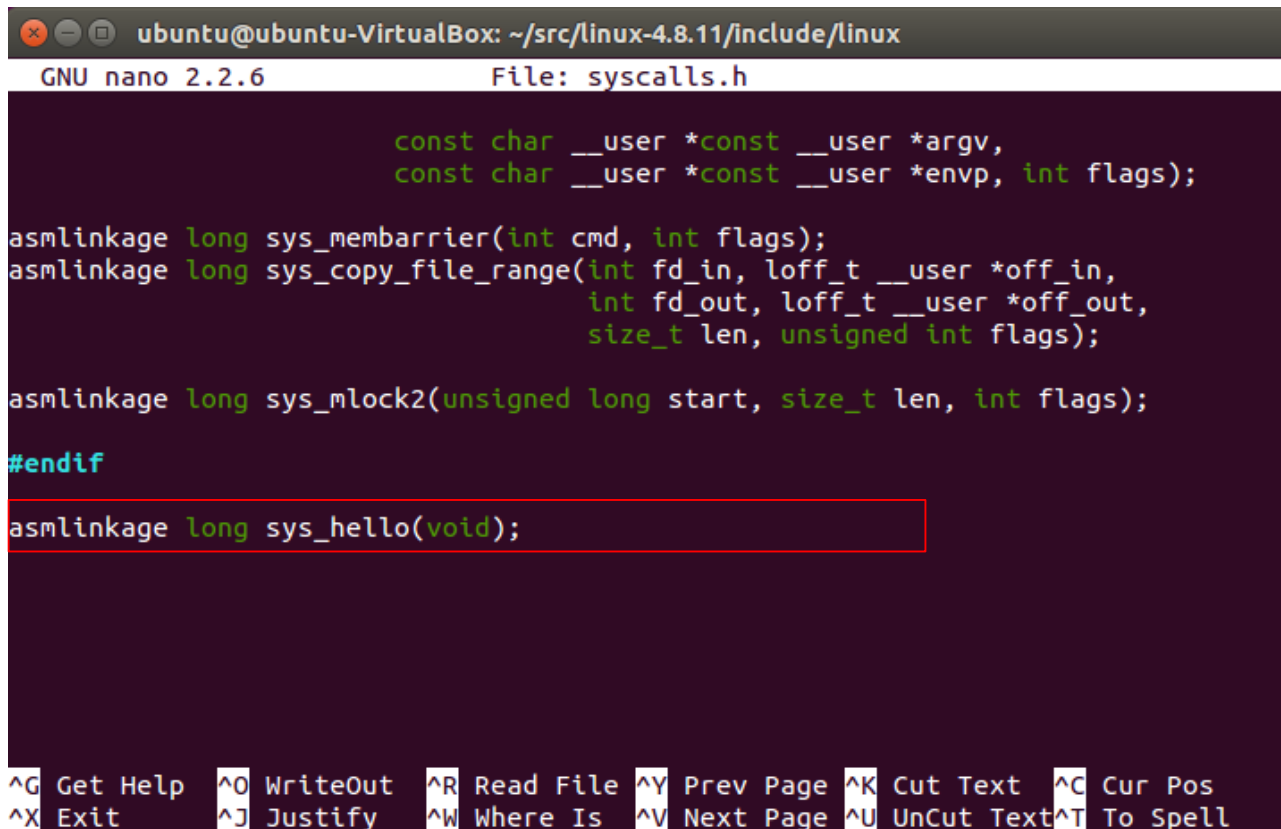
```
ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11/arch/x86/entry/syscalls
GNU nano 2.2.6 File: syscall_32.tbl

372    i386    recvmsg          sys_recvmsg          compat_$
373    i386    shutdown          sys_shutdown
374    i386    userfaultfd       sys_userfaultfd
375    i386    membarrier        sys_membarrier
376    i386    mlock2            sys_mlock2
377    i386    copy_file_range   sys_copy_file_range
378    i386    preadv2           sys_preadv2          compat_$
379    i386    pwritev2          sys_pwritev2         compat_$
380    i386    hello             sys_hello

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

# Dodavanje novog sistemskog poziva

- Dodavanje novog sistemskog poziva u header datoteku sa sistemskim pozivima `~/src/linux-4.8.11/include/linux/syscalls.h`



```
ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11/include/linux
GNU nano 2.2.6 File: syscalls.h

    const char __user *const __user *argv,
    const char __user *const __user *envp, int flags);

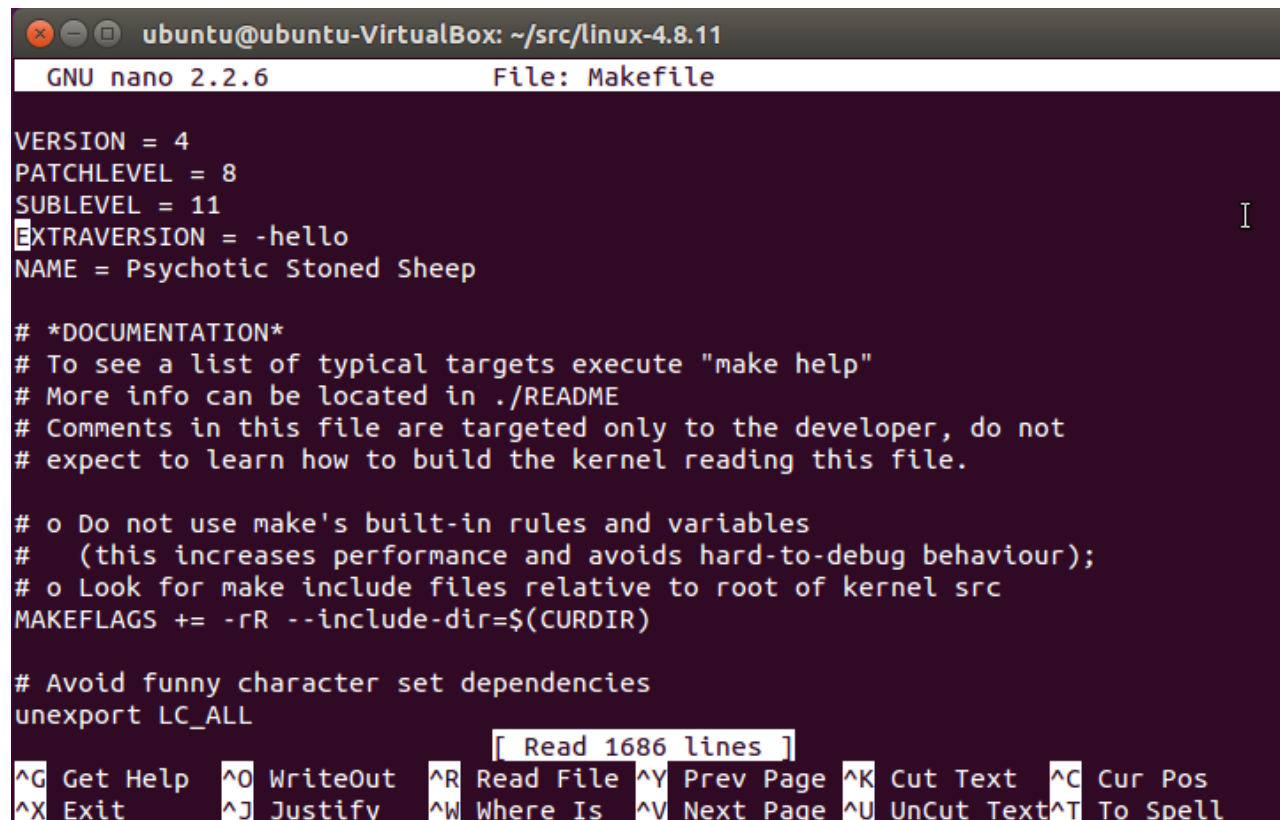
asmlinkage long sys_membarrier(int cmd, int flags);
asmlinkage long sys_copy_file_range(int fd_in, loff_t __user *off_in,
                                     int fd_out, loff_t __user *off_out,
                                     size_t len, unsigned int flags);

asmlinkage long sys_mlock2(unsigned long start, size_t len, int flags);
#endif
asmlinkage long sys_hello(void);
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

# Dodavanje novog sistemskog poziva

- Izmena make datoteke kernela (dodatna oznaka verzije kernela koja se prevodi) ~/src/linux-4.8.11/Makefile



```
ubuntu@ubuntu-VirtualBox: ~/src/linux-4.8.11
GNU nano 2.2.6 File: Makefile

VERSION = 4
PATCHLEVEL = 8
SUBLEVEL = 11
EXTRAVERSION = -hello
NAME = Psychotic Stoned Sheep

# *DOCUMENTATION*
# To see a list of typical targets execute "make help"
# More info can be located in ./README
# Comments in this file are targeted only to the developer, do not
# expect to learn how to build the kernel reading this file.

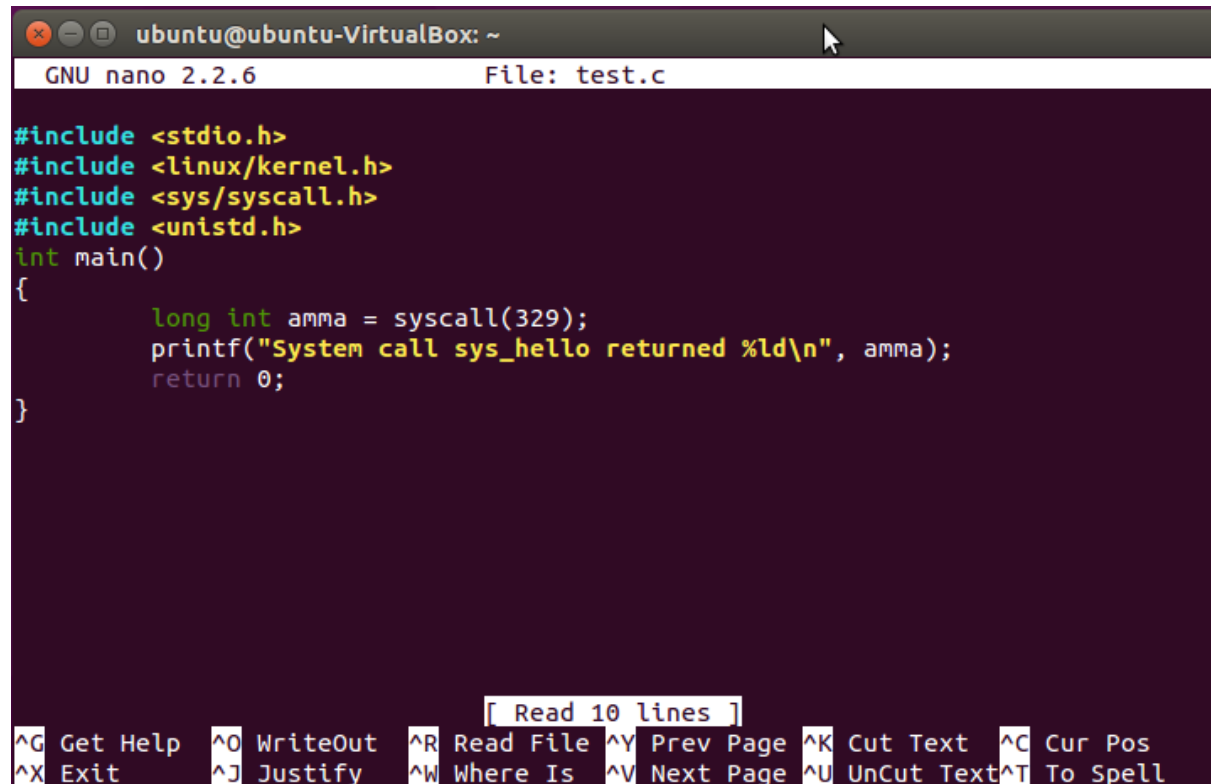
# o Do not use make's built-in rules and variables
#   (this increases performance and avoids hard-to-debug behaviour);
# o Look for make include files relative to root of kernel src
MAKEFLAGS += -rR --include-dir=$(CURDIR)

# Avoid funny character set dependencies
unexport LC_ALL

[ Read 1686 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

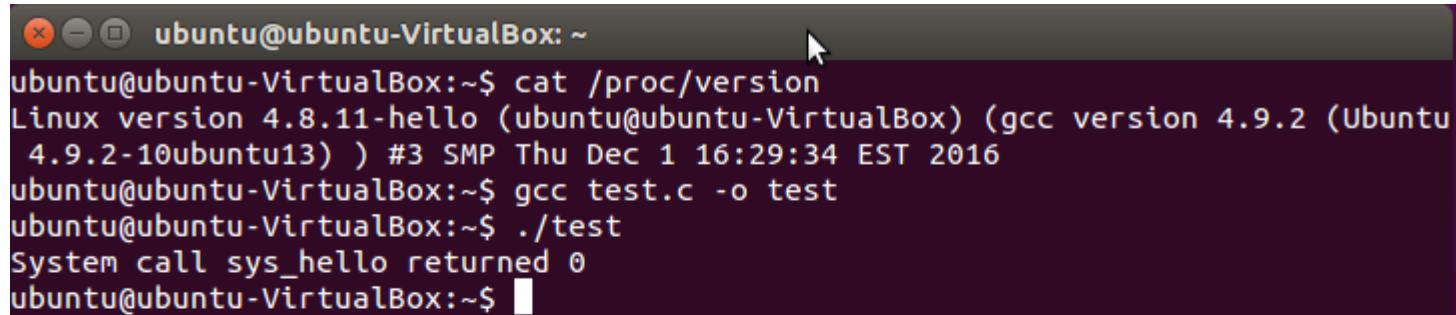
# Testiranje novog sistemskog poziva

- Nakon prevođenja kernela treba testirati novi sistemski poziv
  - Sistem se podiže sa novom verzijom kernela
  - Datoteka test.c može da se nalazi bilo gde u sistemu



```
ubuntu@ubuntu-VirtualBox: ~  
GNU nano 2.2.6 File: test.c  
  
#include <stdio.h>  
#include <linux/kernel.h>  
#include <sys/syscall.h>  
#include <unistd.h>  
int main()  
{  
    long int amma = syscall(329);  
    printf("System call sys_hello returned %ld\n", amma);  
    return 0;  
}  
  
[ Read 10 lines ]  
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos  
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

# Testiranje novog sistemskog poziva

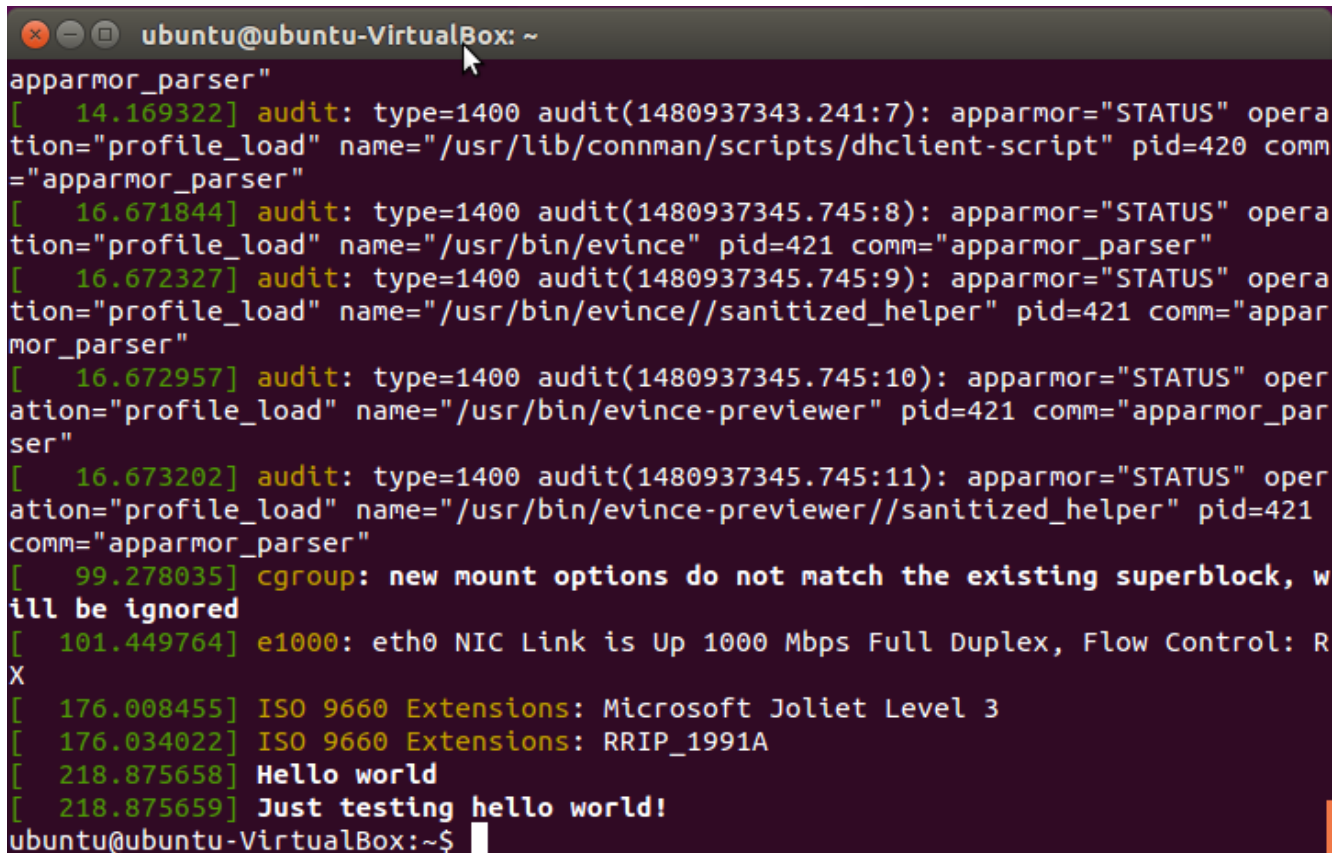
A terminal window titled 'ubuntu@ubuntu-VirtualBox: ~' with standard window controls. The terminal output shows the execution of a system call test. The user runs 'cat /proc/version' which displays the kernel and GCC versions. Then, the user compiles a test program 'test.c' using 'gcc test.c -o test' and runs it with './test'. The output of the program is 'System call sys\_hello returned 0'.

```
ubuntu@ubuntu-VirtualBox: ~  
ubuntu@ubuntu-VirtualBox:~$ cat /proc/version  
Linux version 4.8.11-hello (ubuntu@ubuntu-VirtualBox) (gcc version 4.9.2 (Ubuntu  
4.9.2-10ubuntu13) ) #3 SMP Thu Dec 1 16:29:34 EST 2016  
ubuntu@ubuntu-VirtualBox:~$ gcc test.c -o test  
ubuntu@ubuntu-VirtualBox:~$ ./test  
System call sys_hello returned 0  
ubuntu@ubuntu-VirtualBox:~$
```



# Testiranje novog sistemskog poziva

- Funkcija `printk` štapa poruke u log datoteku kernela. Sadržaj datoteke može da se prikaže korišćenjem komande `dmesg`.



```
ubuntu@ubuntu-VirtualBox: ~  
apparmor_parser"  
[ 14.169322] audit: type=1400 audit(1480937343.241:7): apparmor="STATUS" operation="profile_load" name="/usr/lib/connman/scripts/dhclient-script" pid=420 comm="apparmor_parser"  
[ 16.671844] audit: type=1400 audit(1480937345.745:8): apparmor="STATUS" operation="profile_load" name="/usr/bin/evince" pid=421 comm="apparmor_parser"  
[ 16.672327] audit: type=1400 audit(1480937345.745:9): apparmor="STATUS" operation="profile_load" name="/usr/bin/evince//sanitized_helper" pid=421 comm="apparmor_parser"  
[ 16.672957] audit: type=1400 audit(1480937345.745:10): apparmor="STATUS" operation="profile_load" name="/usr/bin/evince-previewer" pid=421 comm="apparmor_parser"  
[ 16.673202] audit: type=1400 audit(1480937345.745:11): apparmor="STATUS" operation="profile_load" name="/usr/bin/evince-previewer//sanitized_helper" pid=421 comm="apparmor_parser"  
[ 99.278035] cgroup: new mount options do not match the existing superblock, will be ignored  
[ 101.449764] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX  
[ 176.008455] ISO 9660 Extensions: Microsoft Joliet Level 3  
[ 176.034022] ISO 9660 Extensions: RRIP_1991A  
[ 218.875658] Hello world  
[ 218.875659] Just testing hello world!  
ubuntu@ubuntu-VirtualBox:~$
```