Operating Systems Project Report

Project Number (01 / 02 / 03):	01
Name:	蕭望緯
Student ID:	0811521
YouTube link (Format youtube.com/watch?v=[key]):	https://youtu.be/Y0dAnP7n9mw
Date (YYYY-MM-DD):	2021-10-20
Names of the files uploaded to E3:	OS_Project01_0811521.pdf
Physical Machine Total RAM (Example: 8.0 GB):	16GB
Physical Machine CPU (Example: Intel i7-2600K):	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz

Checklist	
Yes/No	**Item**
Yes	**The report name follows the format "OS_ProjectXX_StudentID.pdf".**
Yes	**The report was uploaded to E3 before the deadline.**
Yes	**The YouTube video is public, and anyone with the link can watch it.**
Yes	**The audio of the video has a good volume.**
Yes	**The pictures in your report and video have a good quality.**
Yes	**All the questions and exercises were answered inside the report.**
Yes	**I understand that late submission is late submission, regardless of the time uploaded.**
Yes	**I understand that any cheating in my report / video / code will not be tolerated.**

Individual Questions:

1. What is a Kernel? What are the differences between *mainline, stable* and *longterm*? What is a Kernel panic?

Ans:

Kernel: lies in the center part of an operating system, responsible for process and memory management, file systems, device control and networking.

mainline, stable and *longterm* are three sections that release some kernel versions:

mainline: provide versions that contain new features

stable: versions in the mainline section will be sent to the stable section, and bugfix versions will be released frequently

longterm: bugfix versions are released less frequently

Kernel panic: occurs when any fatal error is detected, and the OS will stop to protect the system

2. What are the differences between *building*, *debugging* and *profiling*?

Ans:

building: install a new kernel

debugging: for the development of kernel

profiling: monitor the performance of processes and resources allocation of the kernel

3. What are GCC, GDB, and KGDB, and what they are used for?

Ans:

GCC: compiler for various programming languages, including C, C++, Fortran, Ada, Go, and etc.

GDB: debugger that aids the development of software programs

KDGB: debugger specific to development of kernel, runs on the Target machine to be debugged

4. What are the /usr/, /boot/, /home/, /boot/grub folders for?

Ans:

/usr: contains important files such as data files, libraries, binaries, etc, which support the system.

/boot: contains files related to the booting of the system

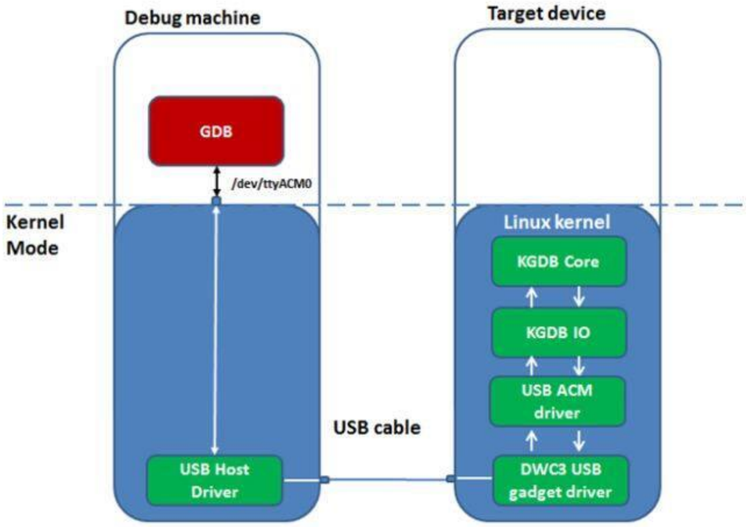
/home: personal workspace for users where the users have full access to writing and creating files

/boot/grub: contains files related to GRUB

5. What are the general steps to debug a Linux Kernel? (Add a figure)

Ans:

- A debugger machine and a target machine
- Connection between them
- gdb on the debugger machine and kgdb on the target machine
- establish debugging session



source: [1]

6. For this project, why do we need two virtual machines?

Ans:

Using kgdb requires two virtual machines. Debugger machine is used as a debugging tool and runs gdb, while Target machine is debugged, runs kgdb, and can be frozen by the debugger. Also, this project simulates remote debugging in real world situation.

7. In Section 3.2, what are the differences between **make**, **make modules_install** and **make install**?

Ans:

make: compile the kernel

make modules_install: install compiled kernel modules

make install: install the kernel so that we can execute it

8. In Section 3.3, what are the commands **kgdbwait** and **kgdboc=ttyS1,115200** for?

Ans:

kgdbwait: Upon booting the kernel, the system will wait for the connection to the KGDB

kgdboc: configure the device through which gdb and kgdb communicate

9. What is **grub**? What is **grub.cfg**?

Ans:

grub: a boot loader that manages the booting process and allows users to launch one of the installed operating system

grub.cfg: GRUB configuration file that contains menu setting and individual setting for each installed kernel. It is generated by several grub.d files.

10. List at least 10 commands you can use with GDB.

Ans:

- connect to remote machine: **target remote /dev/ttyS1**
- unfreeze the target machine: **continue**
- set breakpoint at the beginning of a function: **break** *function*
- list information of all current breakpoints: **info breakpoints**
- delete all breakpoints: **delete**
- delete breakpoints set in the function: **clear** *function*
- print the parameter of the function hitting a breakpoint: **print** *parameter*
- list source code nearby the index: **list *index***
- execute the next line of program (execute the entire function call): **next**
- execute the next line of program (include each line of a function call): **step**
- show the recent locations (several frames, the entire stack) in the program: **backtrace**
- show current stack frame: **frame**
- list information of current frame: **info frame**

11. What is a kernel function? What is a system call?

Ans:

kernel function: jobs supported by the kernel

system call: request services from the kernel

12. What is KASLR? What is it for?

Ans:

"Kernel address space layout randomization"

It enables address space randomization to protect the memory from attack and illegal access.

13. What are GDB's non-stop and all-stop modes?

Ans:

non-stop: in multi-threaded system, GDB freezes certain threads while other threads keep running

all-stop modes: GDB freezes the entire system (all threads)

14. Explain what the command **echo g > /proc/sysrq-trigger** does.

Ans:

stop the kernel execution and give control to kgdb

15. What are these functions: **clone**, **mmap**, **write** and **open?**

Ans:

clone: create a child process

mmap: creates a new mapping of files to memory in the virtual address space of the calling process

write: writes bytes from the buffer to the file pointed by file descriptor.

open: opens the file specified by a pathname.

16. Why is there no **fork** system call? What is the difference between **fork** and **clone?**

Ans:

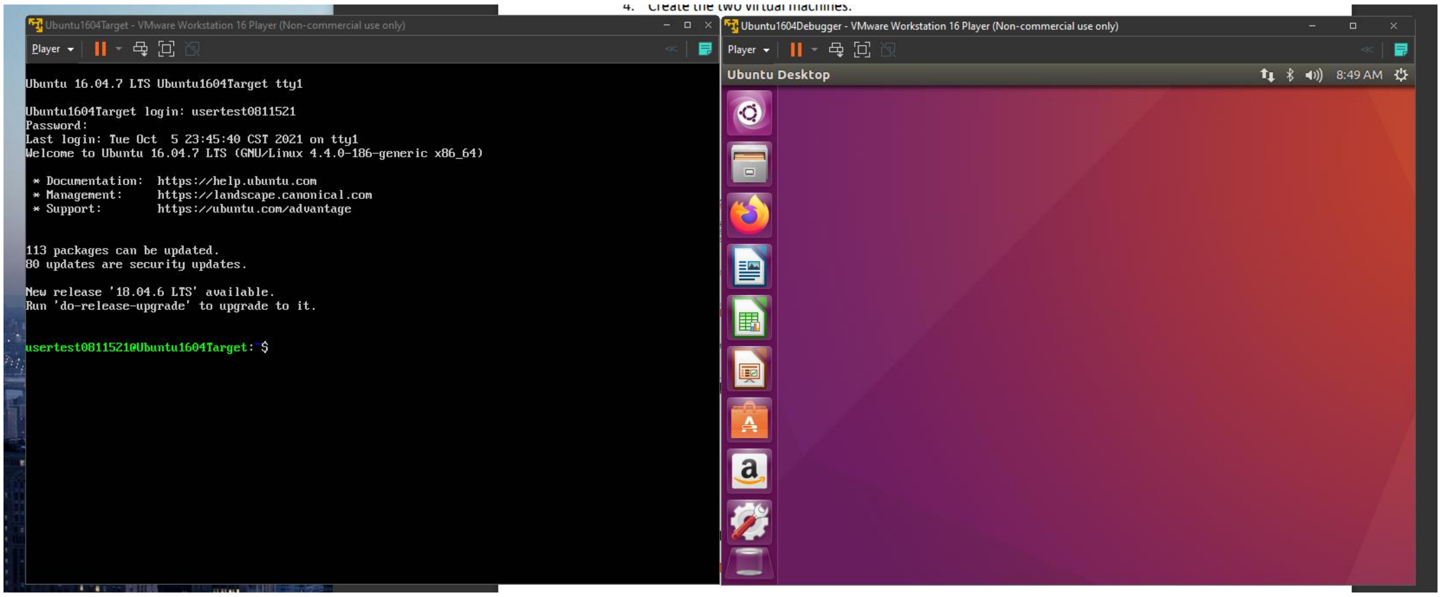
fork will call **clone** system call and do similar things like **clone.**

fork: the created child process doesn't share resources (e.g., memory space, files) with the parent

clone: the created child process can share resources with the parent

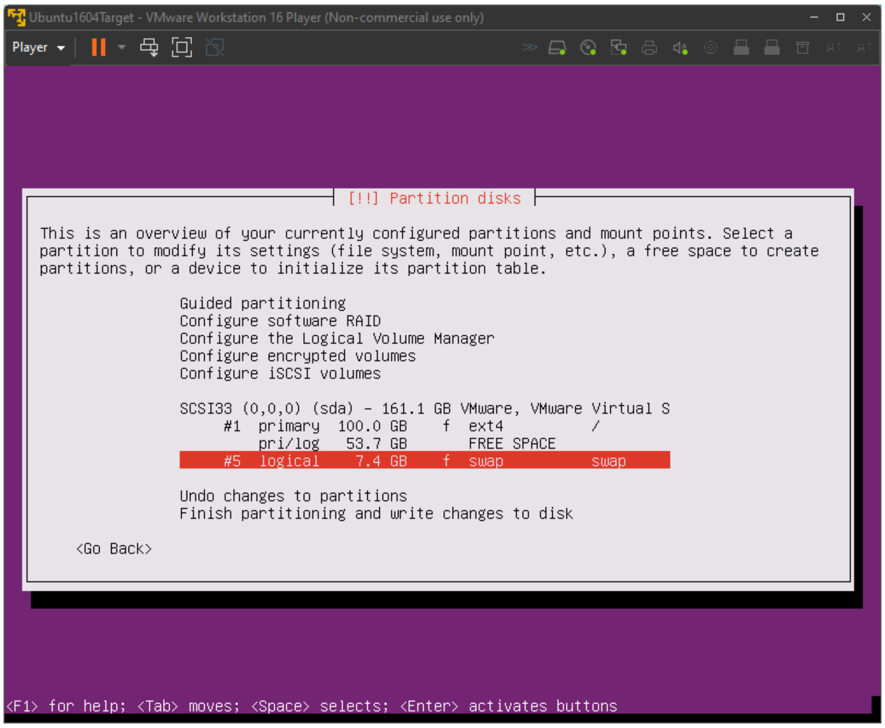
Screenshot #1

Target(left) & Debugger(right) Machine


4. Create the two virtual machines.
Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player
Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player
Ubuntu Desktop
8:49 AM

```
Ubuntu 16.04.7 LTS Ubuntu1604Target tty1

Ubuntu1604Target login: usertest0811521
Password:
Last login: Tue Oct  5 23:45:40 CST 2021 on tty1
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-186-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


113 packages can be updated.
80 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


usertest0811521@Ubuntu1604Target:~$
```

Target machine disks configuration

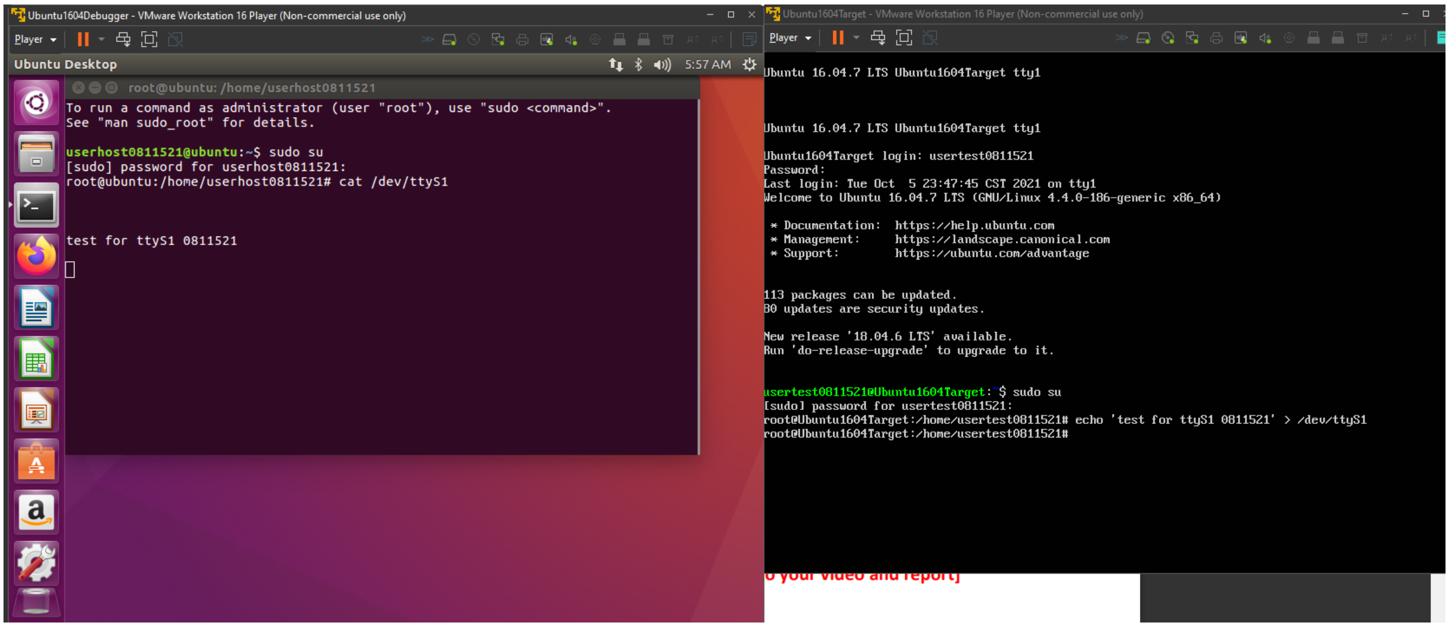

Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player

```
                        [!!] Partition disks

This is an overview of your currently configured partitions and mount points. Select a
partition to modify its settings (file system, mount point, etc.), a free space to create
partitions, or a device to initialize its partition table.

                Guided partitioning
                Configure software RAID
                Configure the Logical Volume Manager
                Configure encrypted volumes
                Configure iSCSI volumes

                SCSI33 (0,0,0) (sda) - 161.1 GB VMware, VMware Virtual S
                     #1  primary  100.0 GB    f  ext4           /
                         pri/log   53.7 GB       FREE SPACE
                     #5  logical    7.4 GB    f  swap           swap

                Undo changes to partitions
                Finish partitioning and write changes to disk

    <Go Back>

<F1> for help; <Tab> moves; <Space> selects; <Enter> activates buttons
```

Screenshot #2

Serial port communication testing after some configuration

Message passed from Target to Debugger successfully through serial port communication



Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player
Ubuntu Desktop
5:57 AM
root@ubuntu: /home/userhost0811521
Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player
o your video and report]

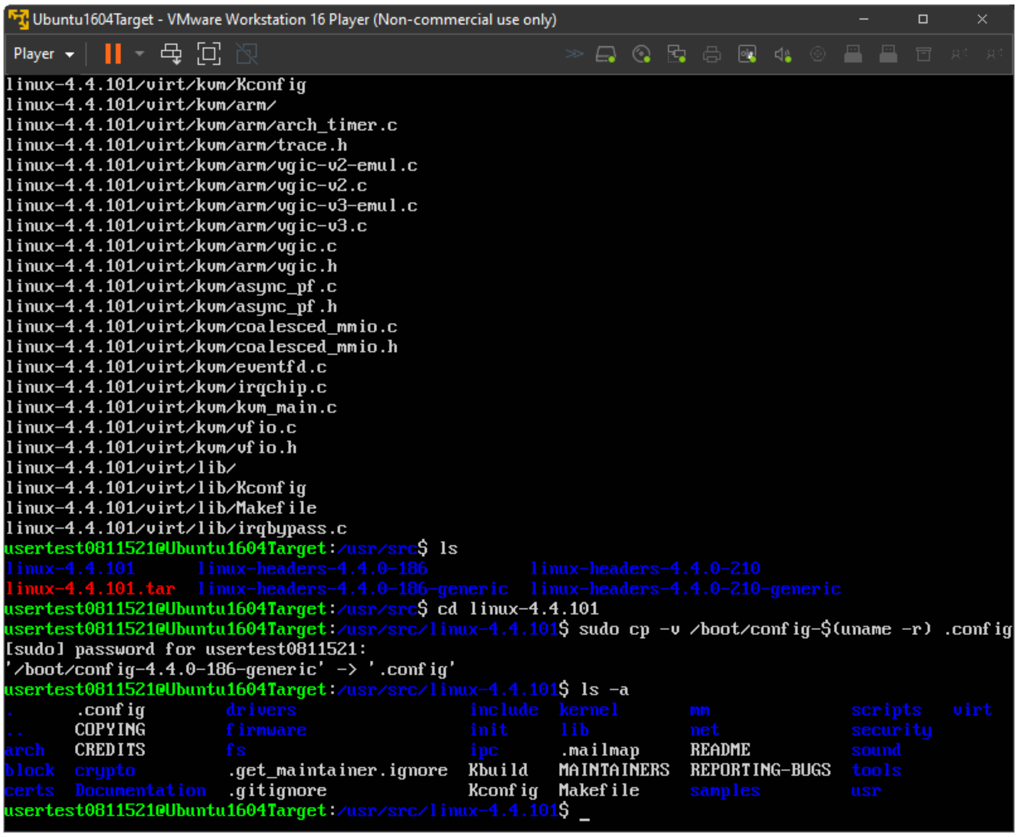
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

userhost0811521@ubuntu:~$ sudo su
[sudo] password for userhost0811521:
root@ubuntu:/home/userhost0811521# cat /dev/ttyS1



test for ttyS1 0811521
```

```
Ubuntu 16.04.7 LTS Ubuntu1604Target tty1



Ubuntu 16.04.7 LTS Ubuntu1604Target tty1

Ubuntu1604Target login: usertest0811521
Password:
Last login: Tue Oct  5 23:47:45 CST 2021 on tty1
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-186-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


113 packages can be updated.
80 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


usertest0811521@Ubuntu1604Target:~$ sudo su
[sudo] password for usertest0811521:
root@Ubuntu1604Target:/home/usertest0811521# echo 'test for ttyS1 0811521' > /dev/ttyS1
root@Ubuntu1604Target:/home/usertest0811521#
```

Screenshot #3

Use the predefined configuration from one of the other versions of kernel

Copy the .config file to the directory of kernel used for this project

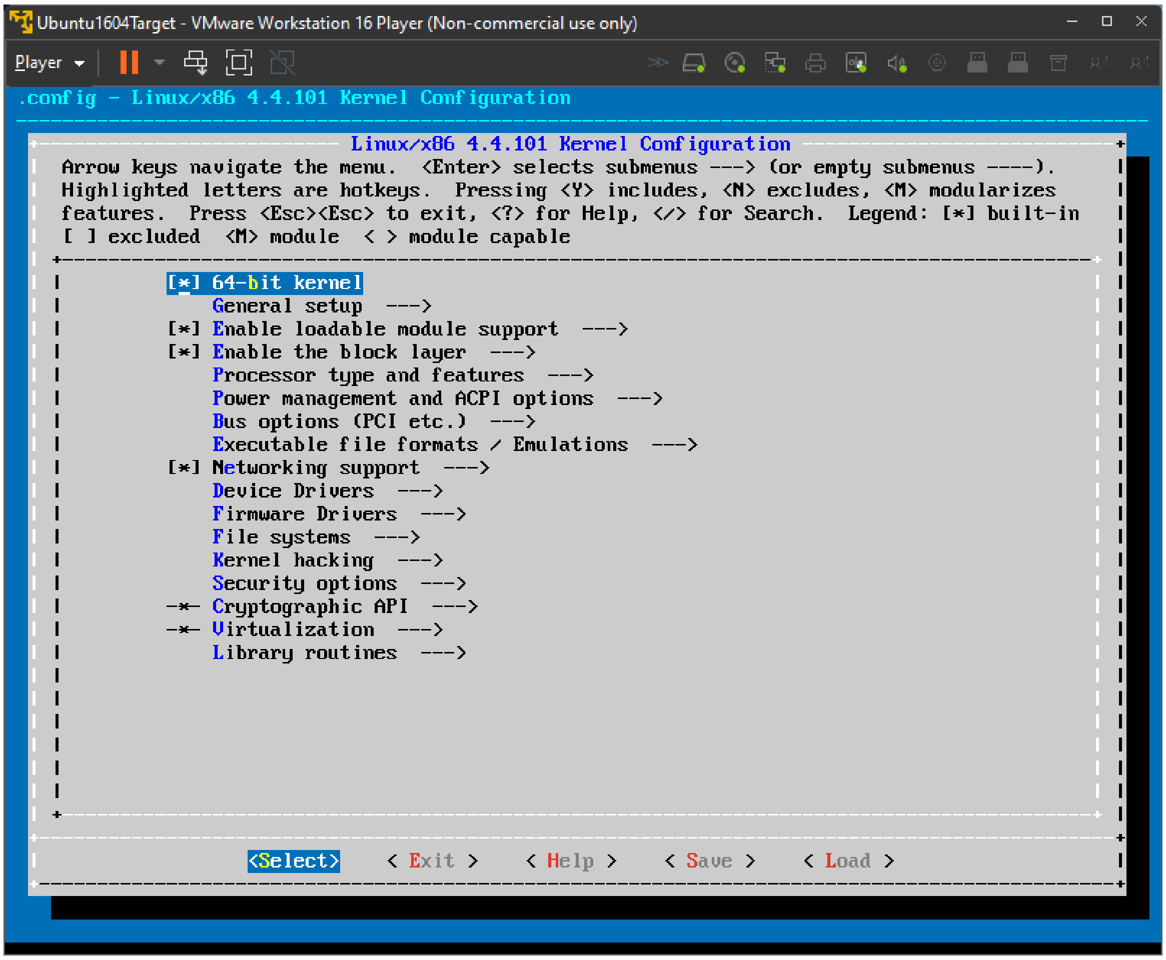


Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player

```
linux-4.4.101/virt/kvm/Kconfig
linux-4.4.101/virt/kvm/arm/
linux-4.4.101/virt/kvm/arm/arch_timer.c
linux-4.4.101/virt/kvm/arm/trace.h
linux-4.4.101/virt/kvm/arm/vgic-v2-emul.c
linux-4.4.101/virt/kvm/arm/vgic-v2.c
linux-4.4.101/virt/kvm/arm/vgic-v3-emul.c
linux-4.4.101/virt/kvm/arm/vgic-v3.c
linux-4.4.101/virt/kvm/arm/vgic.c
linux-4.4.101/virt/kvm/arm/vgic.h
linux-4.4.101/virt/kvm/async_pf.c
linux-4.4.101/virt/kvm/async_pf.h
linux-4.4.101/virt/kvm/coalesced_mmio.c
linux-4.4.101/virt/kvm/coalesced_mmio.h
linux-4.4.101/virt/kvm/eventfd.c
linux-4.4.101/virt/kvm/irqchip.c
linux-4.4.101/virt/kvm/kvm_main.c
linux-4.4.101/virt/kvm/vfio.c
linux-4.4.101/virt/kvm/vfio.h
linux-4.4.101/virt/lib/
linux-4.4.101/virt/lib/Kconfig
linux-4.4.101/virt/lib/Makefile
linux-4.4.101/virt/lib/irqbypass.c
usertest0811521@Ubuntu1604Target:/usr/src$ ls
linux-4.4.101      linux-headers-4.4.0-186          linux-headers-4.4.0-210
linux-4.4.101.tar  linux-headers-4.4.0-186-generic  linux-headers-4.4.0-210-generic
usertest0811521@Ubuntu1604Target:/usr/src$ cd linux-4.4.101
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ sudo cp -v /boot/config-$(uname -r) .config
[sudo] password for usertest0811521:
'/boot/config-4.4.0-186-generic' -> '.config'
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ ls -a
.      .config        drivers         include  kernel       mm              scripts   virt
..     COPYING        firmware        init     lib          net             security
arch   CREDITS        fs              ipc      .mailmap     README          sound
block  crypto         .get_maintainer.ignore  Kbuild   MAINTAINERS  REPORTING-BUGS  tools
certs  Documentation  .gitignore      Kconfig  Makefile     samples         usr
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ _
```

Screenshot #4

Open Kernel configuration interface and examine the setup



Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player

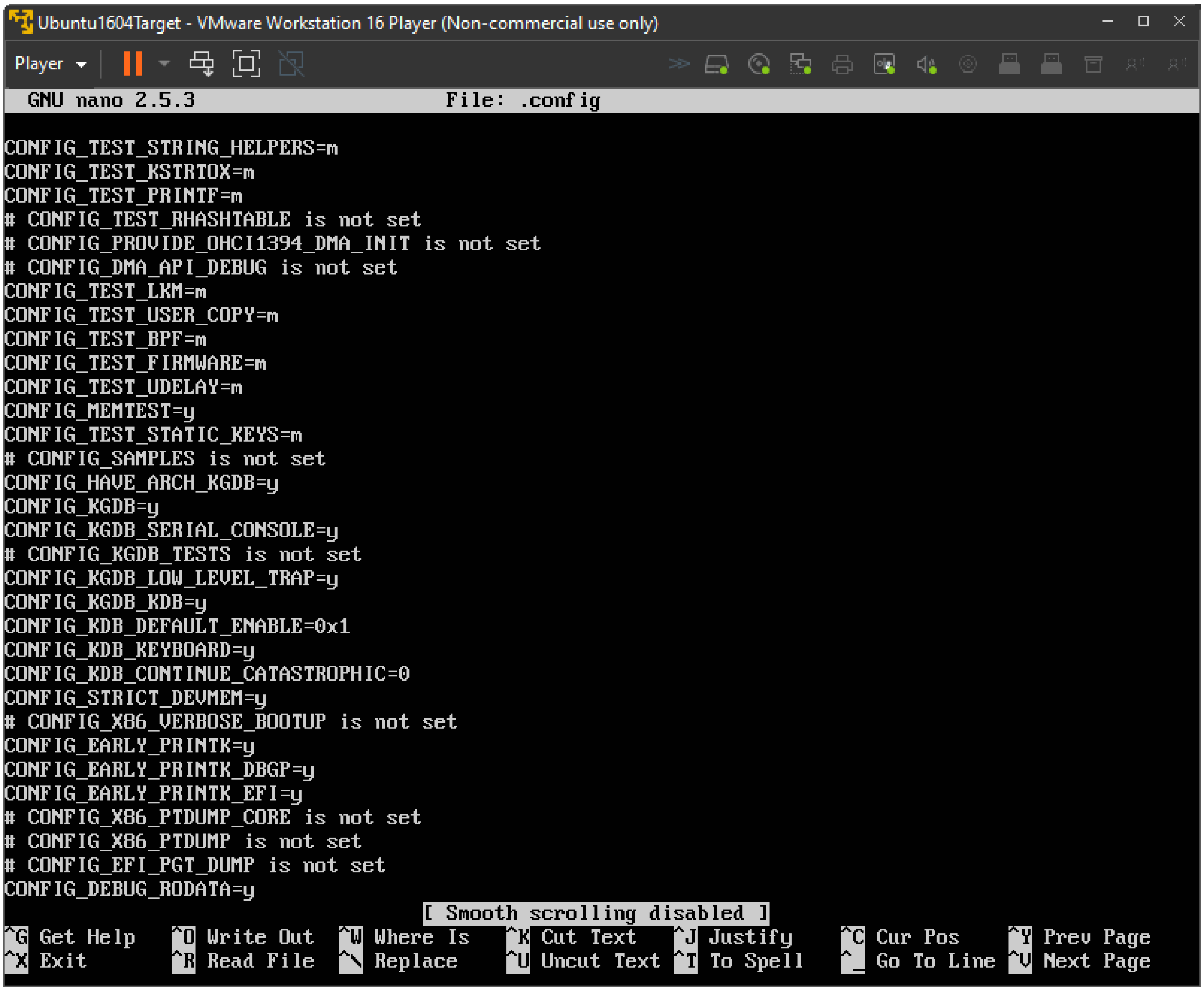
```
.config - Linux/x86 4.4.101 Kernel Configuration

                    Linux/x86 4.4.101 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
  [ ] excluded  <M> module  < > module capable

          [*] 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  --->
          [*] Networking support  --->
              Device Drivers  --->
              Firmware Drivers  --->
              File systems  --->
              Kernel hacking  --->
              Security options  --->
          -*- Cryptographic API  --->
          -*- Virtualization  --->
              Library routines  --->

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

Screenshot #5

Open the file: /usr/src/linux-4.4.101/.config

Double-check these four lines so that we can use kgdb for this kernel in later sections:

```
CONFIG_FRAME_POINTER=y
CONFIG_KGDB=y
CONFIG_KGDB_SERIAL_CONSOLE=y
CONFIG_KGDB_KDB=y
CONFIG_KDB_KEYBOARD=y
```



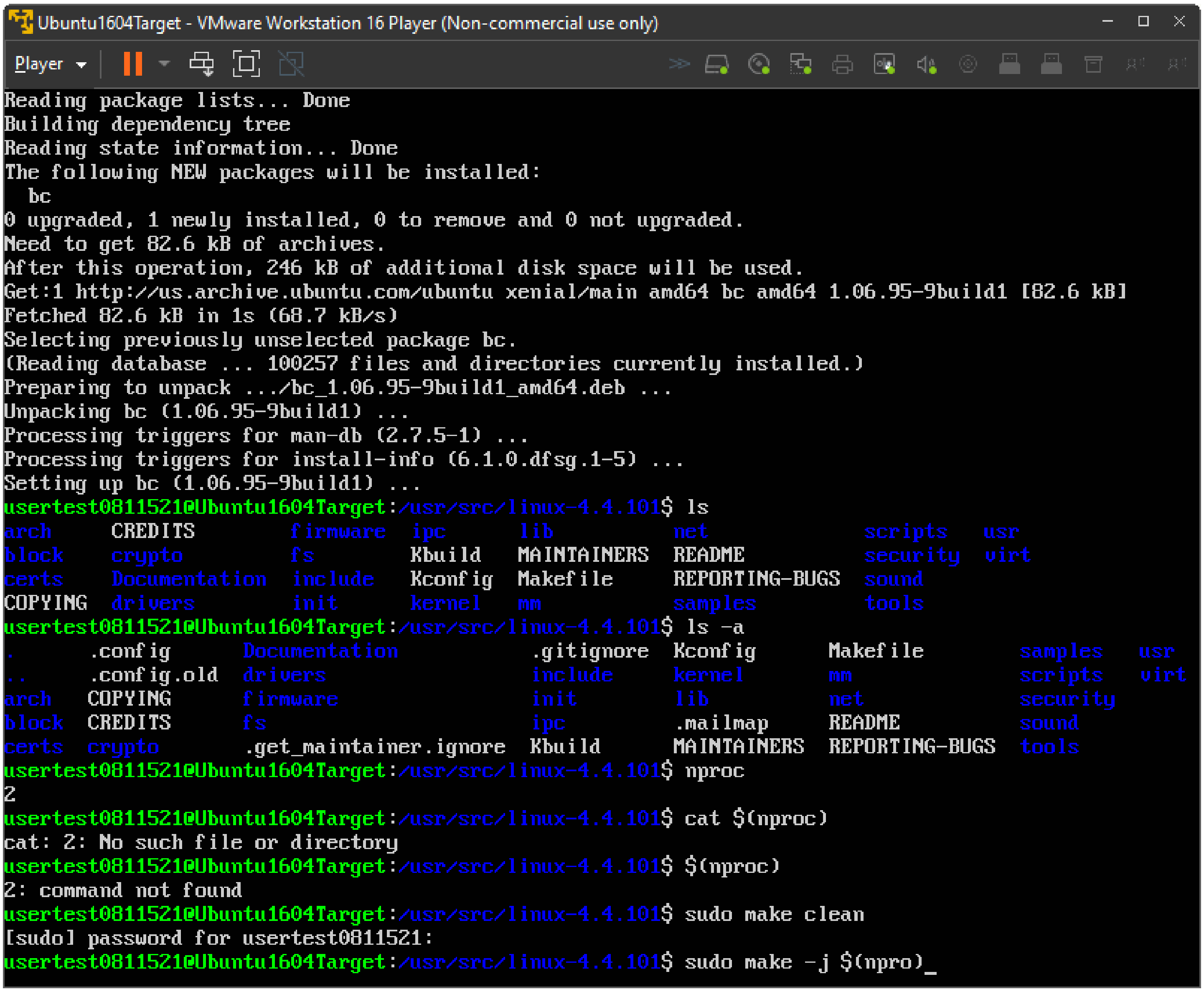
Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player

```
  GNU nano 2.5.3                    File: .config

CONFIG_TEST_STRING_HELPERS=m
CONFIG_TEST_KSTRTOX=m
CONFIG_TEST_PRINTF=m
# CONFIG_TEST_RHASHTABLE is not set
# CONFIG_PROVIDE_OHCI1394_DMA_INIT is not set
# CONFIG_DMA_API_DEBUG is not set
CONFIG_TEST_LKM=m
CONFIG_TEST_USER_COPY=m
CONFIG_TEST_BPF=m
CONFIG_TEST_FIRMWARE=m
CONFIG_TEST_UDELAY=m
CONFIG_MEMTEST=y
CONFIG_TEST_STATIC_KEYS=m
# CONFIG_SAMPLES is not set
CONFIG_HAVE_ARCH_KGDB=y
CONFIG_KGDB=y
CONFIG_KGDB_SERIAL_CONSOLE=y
# CONFIG_KGDB_TESTS is not set
CONFIG_KGDB_LOW_LEVEL_TRAP=y
CONFIG_KGDB_KDB=y
CONFIG_KDB_DEFAULT_ENABLE=0x1
CONFIG_KDB_KEYBOARD=y
CONFIG_KDB_CONTINUE_CATASTROPHIC=0
CONFIG_STRICT_DEVMEM=y
# CONFIG_X86_VERBOSE_BOOTUP is not set
CONFIG_EARLY_PRINTK=y
CONFIG_EARLY_PRINTK_DBGP=y
CONFIG_EARLY_PRINTK_EFI=y
# CONFIG_X86_PTDUMP_CORE is not set
# CONFIG_X86_PTDUMP is not set
# CONFIG_EFI_PGT_DUMP is not set
CONFIG_DEBUG_RODATA=y
                                   [ Smooth scrolling disabled ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  ^V Next Page
```

Screenshot #6

Start building the kernel.

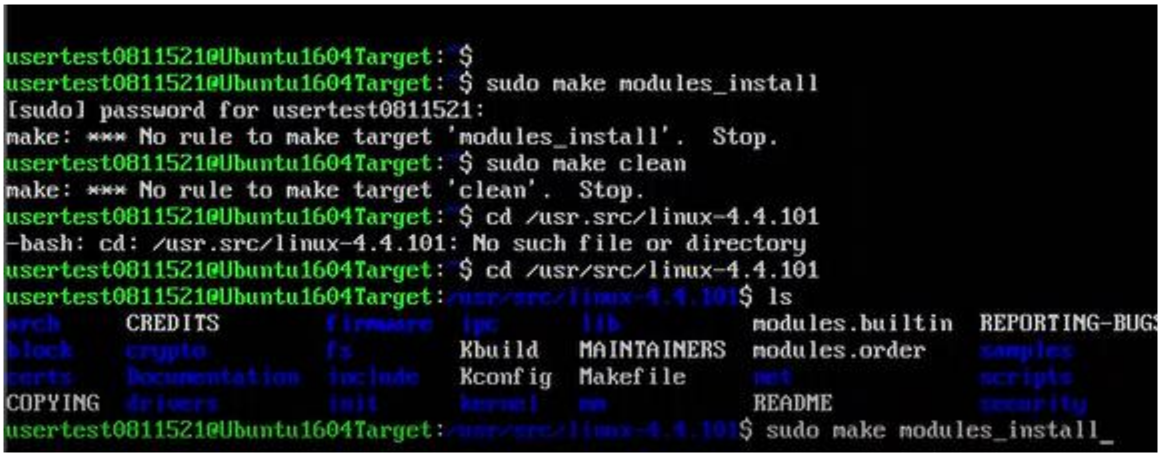
First, clean up the executable or complied object files. Then, build kernel with the maximum available processors.


Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  bc
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 82.6 kB of archives.
After this operation, 246 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 bc amd64 1.06.95-9build1 [82.6 kB]
Fetched 82.6 kB in 1s (68.7 kB/s)
Selecting previously unselected package bc.
(Reading database ... 100257 files and directories currently installed.)
Preparing to unpack .../bc_1.06.95-9build1_amd64.deb ...
Unpacking bc (1.06.95-9build1) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for install-info (6.1.0.dfsg.1-5) ...
Setting up bc (1.06.95-9build1) ...
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ ls
arch     CREDITS        firmware  ipc      lib          net             scripts   usr
block    crypto         fs        Kbuild   MAINTAINERS  README          security  virt
certs    Documentation  include   Kconfig  Makefile     REPORTING-BUGS  sound
COPYING  drivers        init      kernel   mm           samples         tools
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ ls -a
.      .config      Documentation           .gitignore  Kconfig      Makefile        samples   usr
..     .config.old  drivers                 include     kernel       mm              scripts   virt
arch   COPYING      firmware                init        lib          net             security
block  CREDITS      fs                      ipc         .mailmap     README          sound
certs  crypto       .get_maintainer.ignore  Kbuild      MAINTAINERS  REPORTING-BUGS  tools
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ nproc
2
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ cat $(nproc)
cat: 2: No such file or directory
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ $(nproc)
2: command not found
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ sudo make clean
[sudo] password for usertest0811521:
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ sudo make -j $(npro)_
```

Screenshot #7

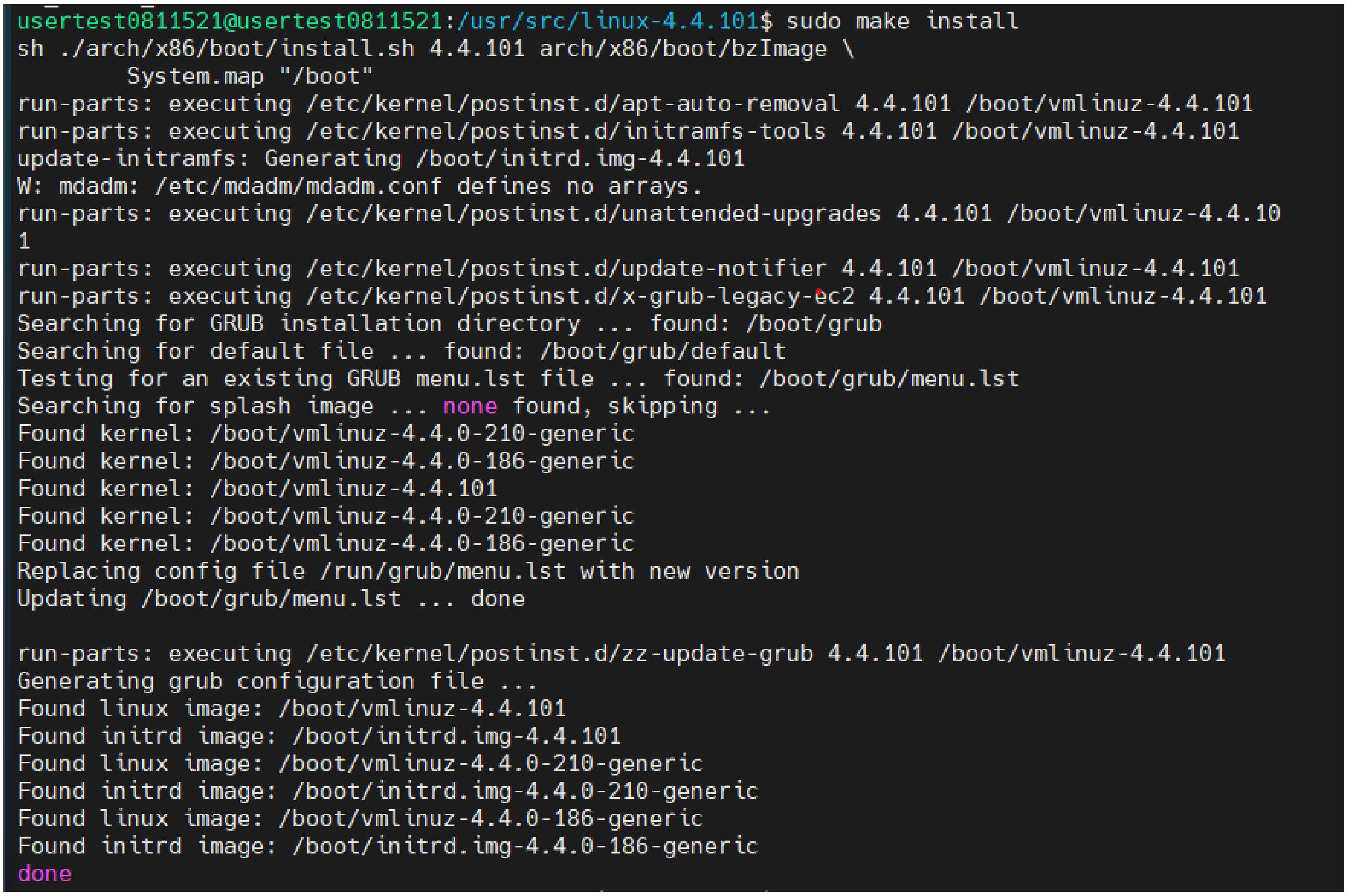
After kernel is compiled, install kernel modules



```
usertest0811521@Ubuntu1604Target: $
usertest0811521@Ubuntu1604Target: $ sudo make modules_install
[sudo] password for usertest0811521:
make: *** No rule to make target 'modules_install'.  Stop.
usertest0811521@Ubuntu1604Target: $ sudo make clean
make: *** No rule to make target 'clean'.  Stop.
usertest0811521@Ubuntu1604Target: $ cd /usr.src/linux-4.4.101
-bash: cd: /usr.src/linux-4.4.101: No such file or directory
usertest0811521@Ubuntu1604Target: $ cd /usr/src/linux-4.4.101
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ ls
arch     CREDITS        firmware  ipc      lib          modules.builtin  REPORTING-BUGS
block    crypto         fs        Kbuild   MAINTAINERS  modules.order    samples
certs    Documentation  include   Kconfig  Makefile     net              scripts
COPYING  drivers        init      kernel   mm           README           security
usertest0811521@Ubuntu1604Target:/usr/src/linux-4.4.101$ sudo make modules_install_
```

Screenshot #8

Install the complete kernel to the system



```
usertest0811521@usertest0811521:/usr/src/linux-4.4.101$ sudo make install
sh ./arch/x86/boot/install.sh 4.4.101 arch/x86/boot/bzImage \
        System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.4.101 /boot/vmlinuz-4.4.101
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.4.101 /boot/vmlinuz-4.4.101
update-initramfs: Generating /boot/initrd.img-4.4.101
W: mdadm: /etc/mdadm/mdadm.conf defines no arrays.
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.4.101 /boot/vmlinuz-4.4.10
1
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.4.101 /boot/vmlinuz-4.4.101
run-parts: executing /etc/kernel/postinst.d/x-grub-legacy-ec2 4.4.101 /boot/vmlinuz-4.4.101
Searching for GRUB installation directory ... found: /boot/grub
Searching for default file ... found: /boot/grub/default
Testing for an existing GRUB menu.lst file ... found: /boot/grub/menu.lst
Searching for splash image ... none found, skipping ...
Found kernel: /boot/vmlinuz-4.4.0-210-generic
Found kernel: /boot/vmlinuz-4.4.0-186-generic
Found kernel: /boot/vmlinuz-4.4.101
Found kernel: /boot/vmlinuz-4.4.0-210-generic
Found kernel: /boot/vmlinuz-4.4.0-186-generic
Replacing config file /run/grub/menu.lst with new version
Updating /boot/grub/menu.lst ... done

run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.4.101 /boot/vmlinuz-4.4.101
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.4.101
Found initrd image: /boot/initrd.img-4.4.101
Found linux image: /boot/vmlinuz-4.4.0-210-generic
Found initrd image: /boot/initrd.img-4.4.0-210-generic
Found linux image: /boot/vmlinuz-4.4.0-186-generic
Found initrd image: /boot/initrd.img-4.4.0-186-generic
done
```

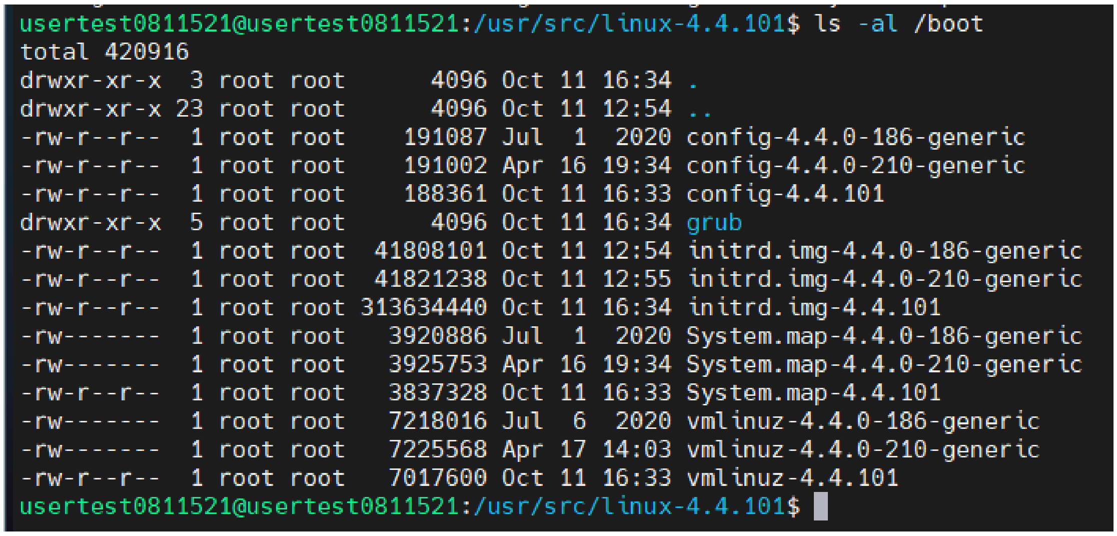
In the /boot directory, the four files of the kernel 4.4.101 are present.

a. initrd.img-4.4.101

b. vmlinuz-4.4.101

c. system.map.4.4.101

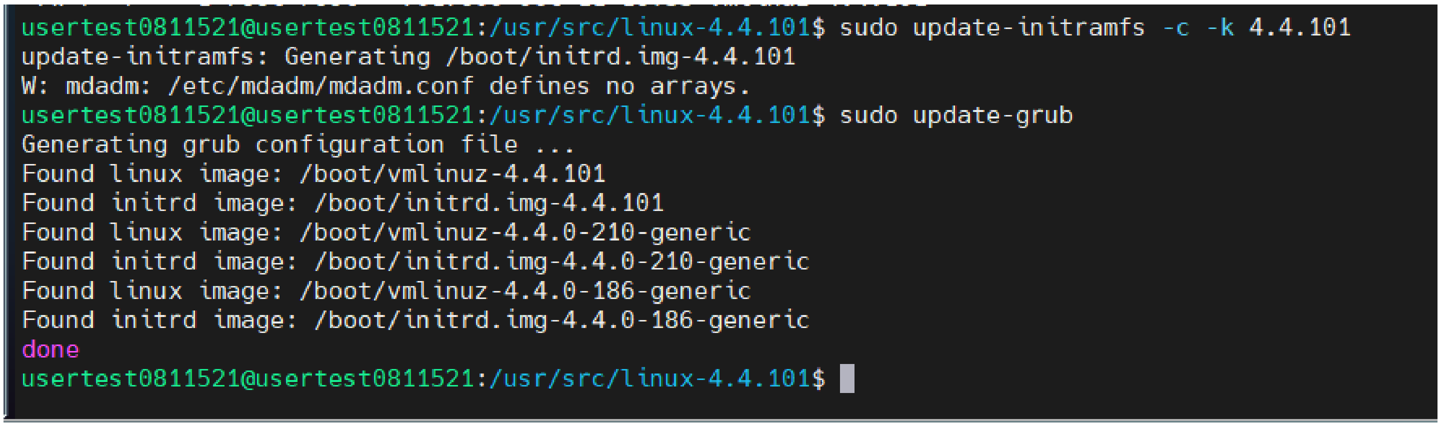
d. config-4.4.101



```
usertest0811521@usertest0811521:/usr/src/linux-4.4.101$ ls -al /boot
total 420916
drwxr-xr-x  3 root root      4096 Oct 11 16:34 .
drwxr-xr-x 23 root root      4096 Oct 11 12:54 ..
-rw-r--r--  1 root root    191087 Jul  1  2020 config-4.4.0-186-generic
-rw-r--r--  1 root root    191002 Apr 16 19:34 config-4.4.0-210-generic
-rw-r--r--  1 root root    188361 Oct 11 16:33 config-4.4.101
drwxr-xr-x  5 root root      4096 Oct 11 16:34 grub
-rw-r--r--  1 root root  41808101 Oct 11 12:54 initrd.img-4.4.0-186-generic
-rw-r--r--  1 root root  41821238 Oct 11 12:55 initrd.img-4.4.0-210-generic
-rw-r--r--  1 root root 313634440 Oct 11 16:34 initrd.img-4.4.101
-rw-------  1 root root   3920886 Jul  1  2020 System.map-4.4.0-186-generic
-rw-------  1 root root   3925753 Apr 16 19:34 System.map-4.4.0-210-generic
-rw-r--r--  1 root root   3837328 Oct 11 16:33 System.map-4.4.101
-rw-------  1 root root   7218016 Jul  6  2020 vmlinuz-4.4.0-186-generic
-rw-------  1 root root   7225568 Apr 17 14:03 vmlinuz-4.4.0-210-generic
-rw-r--r--  1 root root   7017600 Oct 11 16:33 vmlinuz-4.4.101
usertest0811521@usertest0811521:/usr/src/linux-4.4.101$
```

Screenshot #9

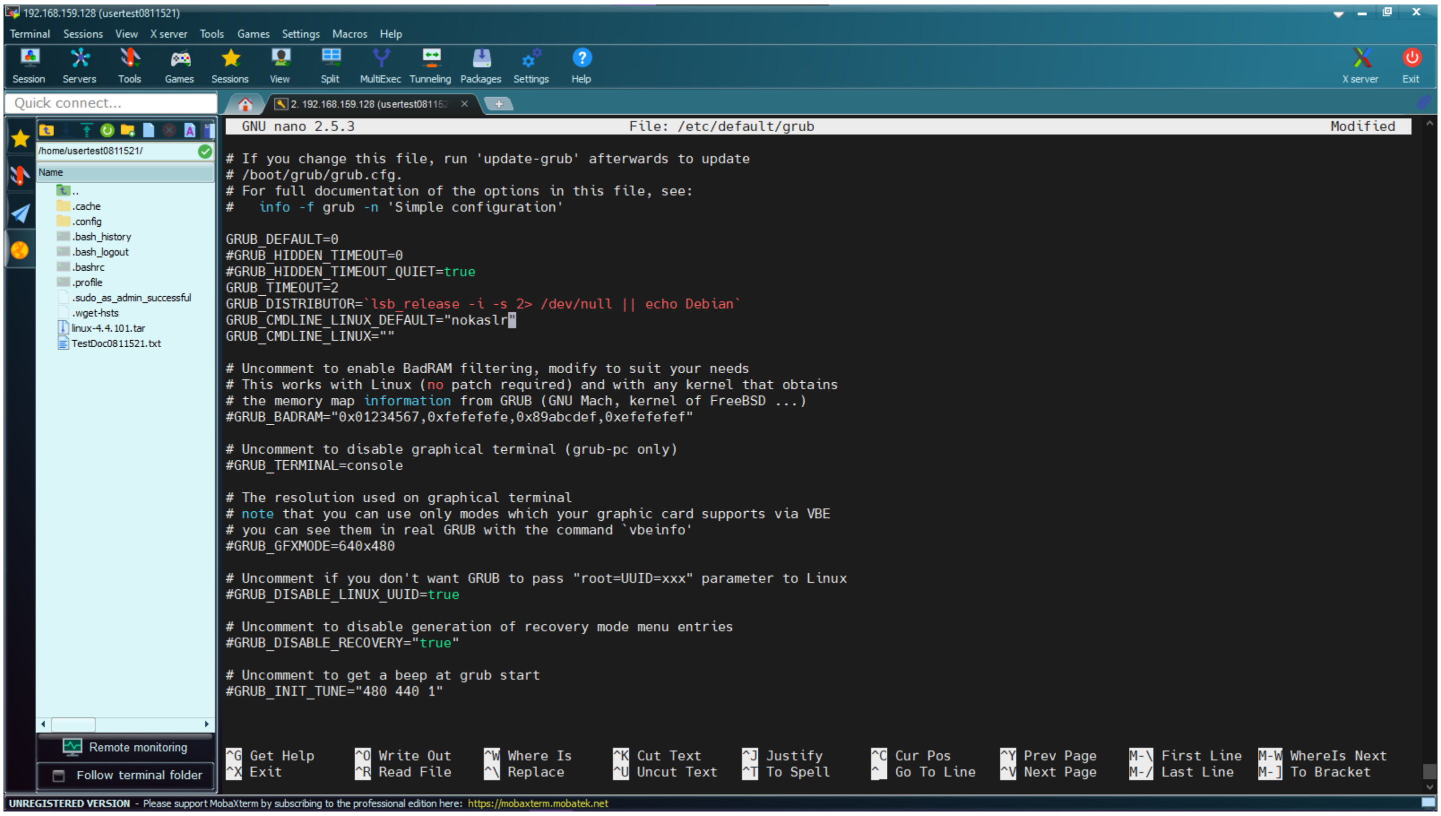
update initrd.img. Then, update GRUB



```
usertest0811521@usertest0811521:/usr/src/linux-4.4.101$ sudo update-initramfs -c -k 4.4.101
update-initramfs: Generating /boot/initrd.img-4.4.101
W: mdadm: /etc/mdadm/mdadm.conf defines no arrays.
usertest0811521@usertest0811521:/usr/src/linux-4.4.101$ sudo update-grub
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.4.101
Found initrd image: /boot/initrd.img-4.4.101
Found linux image: /boot/vmlinuz-4.4.0-210-generic
Found initrd image: /boot/initrd.img-4.4.0-210-generic
Found linux image: /boot/vmlinuz-4.4.0-186-generic
Found initrd image: /boot/initrd.img-4.4.0-186-generic
done
usertest0811521@usertest0811521:/usr/src/linux-4.4.101$
```

Screenshot #10

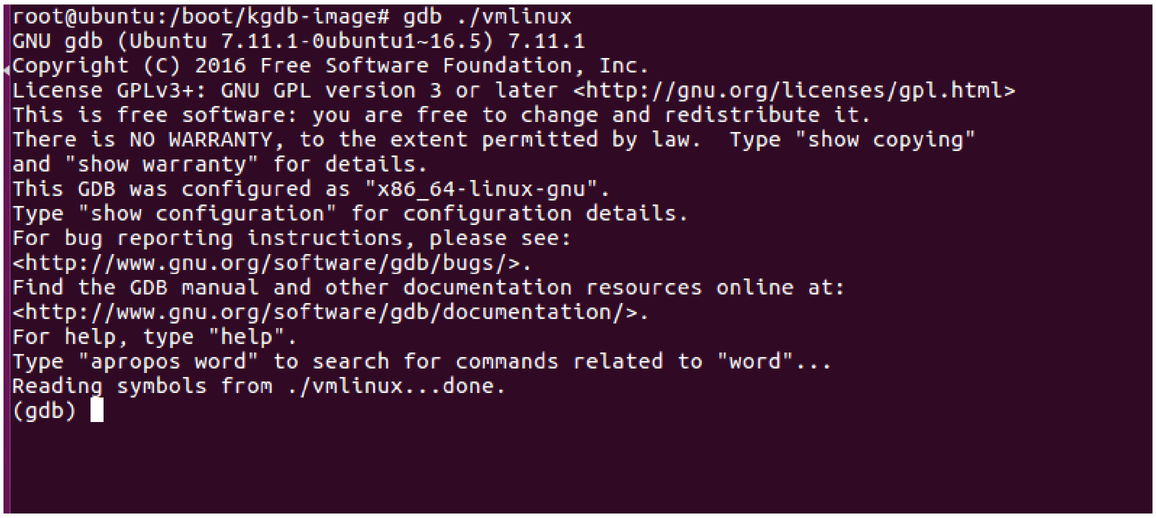
set up parameters for GRUB so that we can see the menu for different kernels and set nokaslr. Then, update GRUB


192.168.159.128 (usertest0811521)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. 192.168.159.128 (usertest081152
/home/usertest0811521/
Name
..
.cache
.config
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.wget-hsts
linux-4.4.101.tar
TestDoc0811521.txt
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```
  GNU nano 2.5.3                          File: /etc/default/grub                                    Modified

# 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_HIDDEN_TIMEOUT=0
#GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=2
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="nokaslr"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page   M-\ First Line M-W WhereIs Next
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  ^V Next Page   M-/ Last Line  M-] To Bracket
```

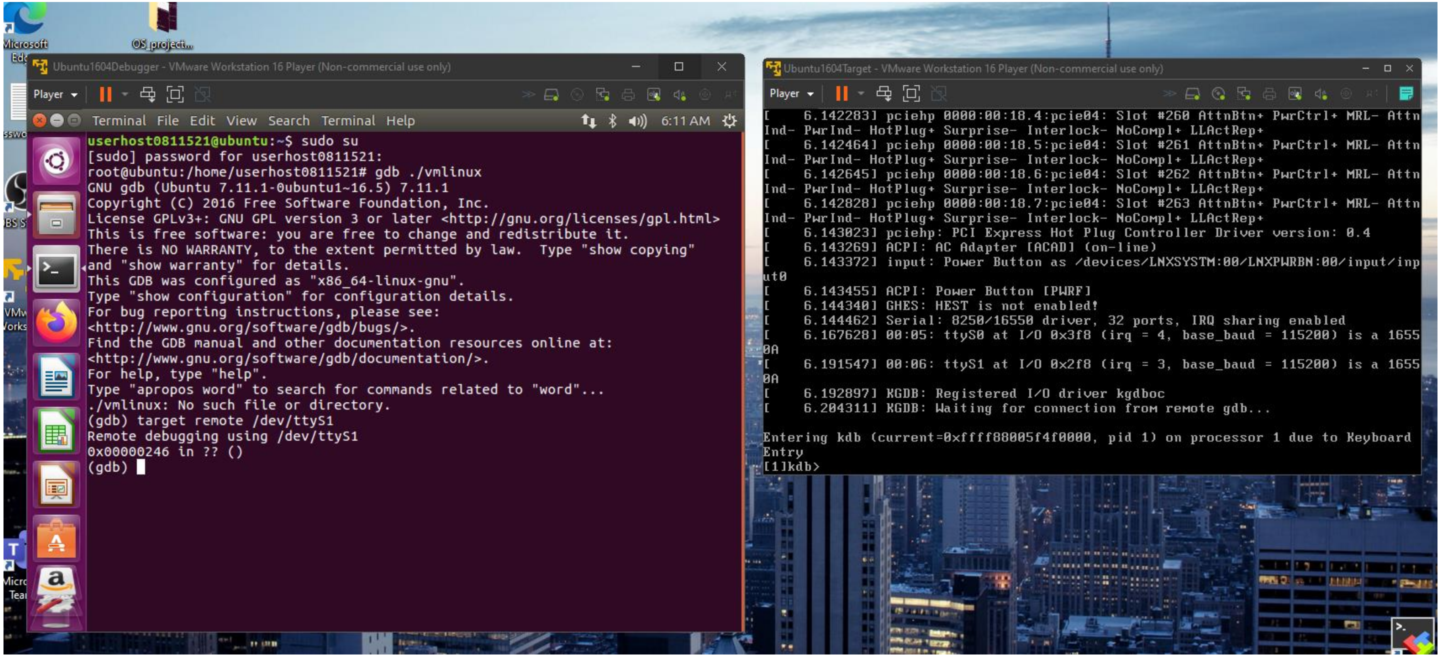
Screenshot #11

gdb ./vmlinux: open gdb



```
root@ubuntu:/boot/kgdb-image# gdb ./vmlinux
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./vmlinux...done.
(gdb) 
```

target remote /dev/ttyS1: connect to remote target machine



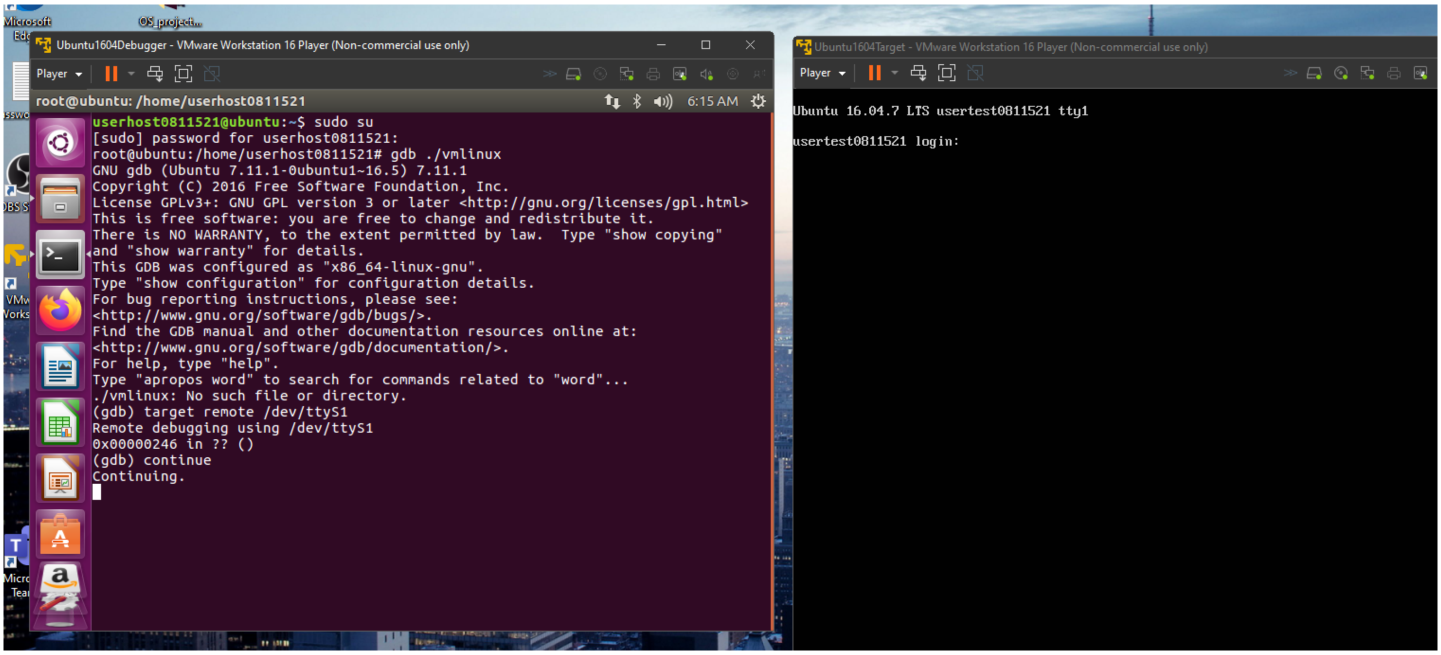
Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player
Terminal File Edit View Search Terminal Help
6:11 AM
Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player

```
userhost0811521@ubuntu:~$ sudo su
[sudo] password for userhost0811521:
root@ubuntu:/home/userhost0811521# gdb ./vmlinux
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
./vmlinux: No such file or directory.
(gdb) target remote /dev/ttyS1
Remote debugging using /dev/ttyS1
0x00000246 in ?? ()
(gdb) 
```

```
[    6.142283] pciehp 0000:00:18.4:pcie04: Slot #260 AttnBtn+ PwrCtrl+ MRL- Attn
Ind- PwrInd- HotPlug+ Surprise- Interlock- NoCompl+ LLActRep+
[    6.142464] pciehp 0000:00:18.5:pcie04: Slot #261 AttnBtn+ PwrCtrl+ MRL- Attn
Ind- PwrInd- HotPlug+ Surprise- Interlock- NoCompl+ LLActRep+
[    6.142645] pciehp 0000:00:18.6:pcie04: Slot #262 AttnBtn+ PwrCtrl+ MRL- Attn
Ind- PwrInd- HotPlug+ Surprise- Interlock- NoCompl+ LLActRep+
[    6.142828] pciehp 0000:00:18.7:pcie04: Slot #263 AttnBtn+ PwrCtrl+ MRL- Attn
Ind- PwrInd- HotPlug+ Surprise- Interlock- NoCompl+ LLActRep+
[    6.143023] pciehp: PCI Express Hot Plug Controller Driver version: 0.4
[    6.143269] ACPI: AC Adapter [ACAD] (on-line)
[    6.143372] input: Power Button as /devices/LNXSYSTM:00/LNXPWRBN:00/input/inp
ut0
[    6.143455] ACPI: Power Button [PWRF]
[    6.144340] GHES: HEST is not enabled!
[    6.144462] Serial: 8250/16550 driver, 32 ports, IRQ sharing enabled
[    6.167628] 00:05: ttyS0 at I/O 0x3f8 (irq = 4, base_baud = 115200) is a 1655
0A
[    6.191547] 00:06: ttyS1 at I/O 0x2f8 (irq = 3, base_baud = 115200) is a 1655
0A
[    6.192897] KGDB: Registered I/O driver kgdboc
[    6.204311] KGDB: Waiting for connection from remote gdb...

Entering kdb (current=0xffff88005f4f0000, pid 1) on processor 1 due to Keyboard
Entry
[1]kdb>
```

Screenshot #12

After I enter command **continue** in the debugger machine, the target machine resumes booting its system.



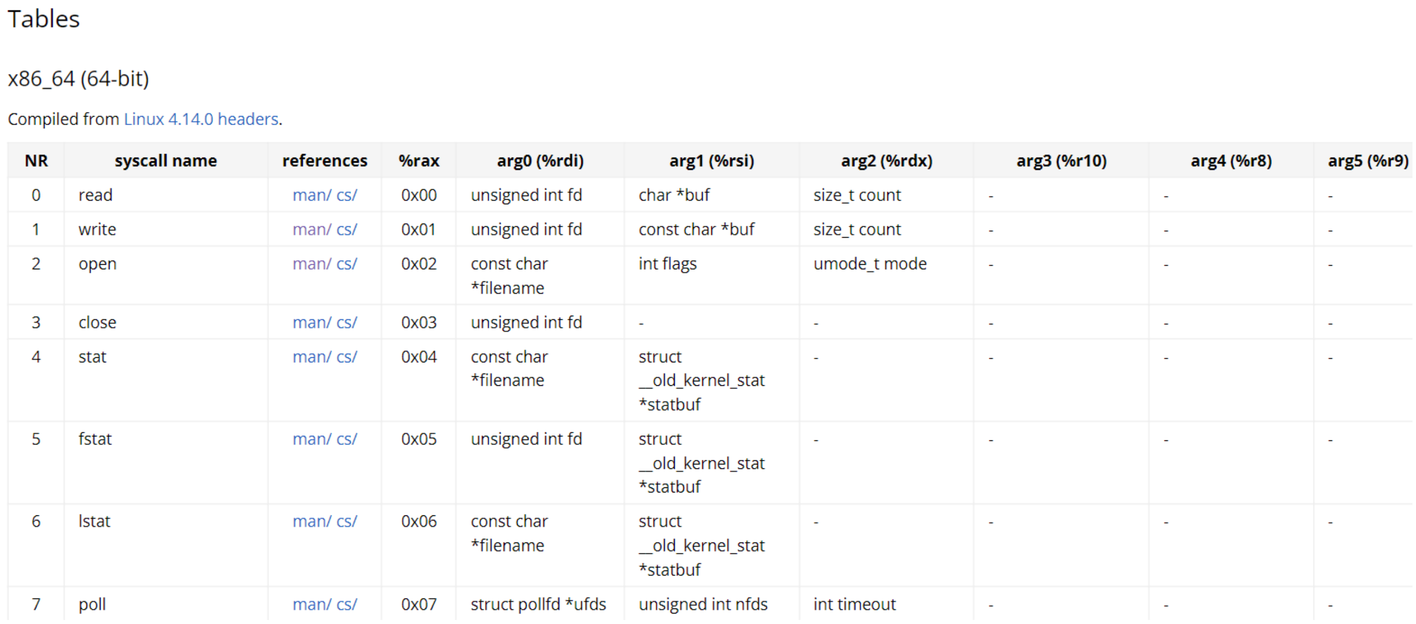
Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player
root@ubuntu: /home/userhost0811521
6:15 AM
Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player

```
userhost0811521@ubuntu:~$ sudo su
[sudo] password for userhost0811521:
root@ubuntu:/home/userhost0811521# gdb ./vmlinux
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
./vmlinux: No such file or directory.
(gdb) target remote /dev/ttyS1
Remote debugging using /dev/ttyS1
0x00000246 in ?? ()
(gdb) continue
Continuing.
```

```
Ubuntu 16.04.7 LTS usertest0811521 tty1

usertest0811521 login:
```

Screenshot #13

Chromium OS Docs - Linux System Call Table (googlesource.com)



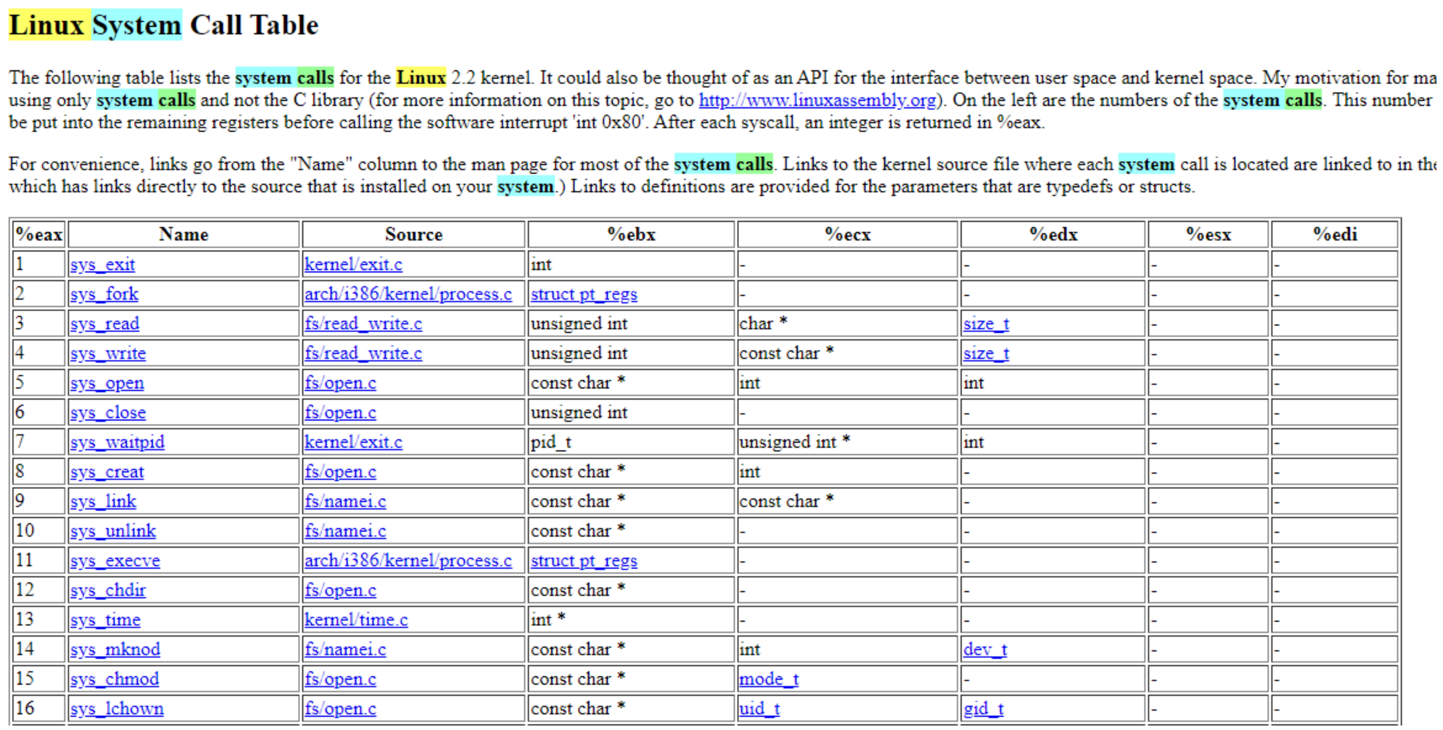
Tables

x86_64 (64-bit)

Compiled from Linux 4.14.0 headers.

NR	syscall name	references	%rax	arg0 (%rdi)	arg1 (%rsi)	arg2 (%rdx)	arg3 (%r10)	arg4 (%r8)	arg5 (%r9)
0	read	man/ cs/	0x00	unsigned int fd	char *buf	size_t count	-	-	-
1	write	man/ cs/	0x01	unsigned int fd	const char *buf	size_t count	-	-	-
2	open	man/ cs/	0x02	const char *filename	int flags	umode_t mode	-	-	-
3	close	man/ cs/	0x03	unsigned int fd	-	-	-	-	-
4	stat	man/ cs/	0x04	const char *filename	struct __old_kernel_stat *statbuf	-	-	-	-
5	fstat	man/ cs/	0x05	unsigned int fd	struct __old_kernel_stat *statbuf	-	-	-	-
6	lstat	man/ cs/	0x06	const char *filename	struct __old_kernel_stat *statbuf	-	-	-	-
7	poll	man/ cs/	0x07	struct pollfd *ufds	unsigned int nfds	int timeout	-	-	-

Linux System Call Table (nps.edu)



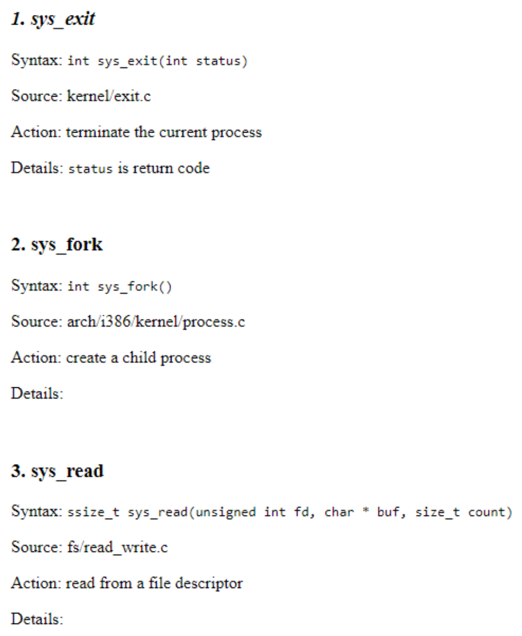
Linux System Call Table

The following table lists the **system calls** for the **Linux** 2.2 kernel. It could also be thought of as an API for the interface between user space and kernel space. My motivation for ma using only **system calls** and not the C library (for more information on this topic, go to http://www.linuxassembly.org). On the left are the numbers of the **system calls**. This number be put into the remaining registers before calling the software interrupt 'int 0x80'. After each syscall, an integer is returned in %eax.

For convenience, links go from the "Name" column to the man page for most of the **system calls**. Links to the kernel source file where each **system** call is located are linked to in the which has links directly to the source that is installed on your **system**.) Links to definitions are provided for the parameters that are typedefs or structs.

%eax	Name	Source	%ebx	%ecx	%edx	%esx	%edi
1	sys_exit	kernel/exit.c	int	-	-	-	-
2	sys_fork	arch/i386/kernel/process.c	struct pt_regs	-	-	-	-
3	sys_read	fs/read_write.c	unsigned int	char *	size_t	-	-
4	sys_write	fs/read_write.c	unsigned int	const char *	size_t	-	-
5	sys_open	fs/open.c	const char *	int	int	-	-
6	sys_close	fs/open.c	unsigned int	-	-	-	-
7	sys_waitpid	kernel/exit.c	pid_t	unsigned int *	int	-	-
8	sys_creat	fs/open.c	const char *	int	-	-	-
9	sys_link	fs/namei.c	const char *	const char *	-	-	-
10	sys_unlink	fs/namei.c	const char *	-	-	-	-
11	sys_execve	arch/i386/kernel/process.c	struct pt_regs	-	-	-	-
12	sys_chdir	fs/open.c	const char *	-	-	-	-
13	sys_time	kernel/time.c	int *	-	-	-	-
14	sys_mknod	fs/namei.c	const char *	int	dev_t	-	-
15	sys_chmod	fs/open.c	const char *	mode_t	-	-	-
16	sys_lchown	fs/open.c	const char *	uid_t	gid_t	-	-

Linux/i386 system calls (sourceforge.net)



1. sys_exit

Syntax: `int sys_exit(int status)`

Source: kernel/exit.c

Action: terminate the current process

Details: `status` is return code

2. sys_fork

Syntax: `int sys_fork()`

Source: arch/i386/kernel/process.c

Action: create a child process

Details:

3. sys_read

Syntax: `ssize_t sys_read(unsigned int fd, char * buf, size_t count)`

Source: fs/read_write.c

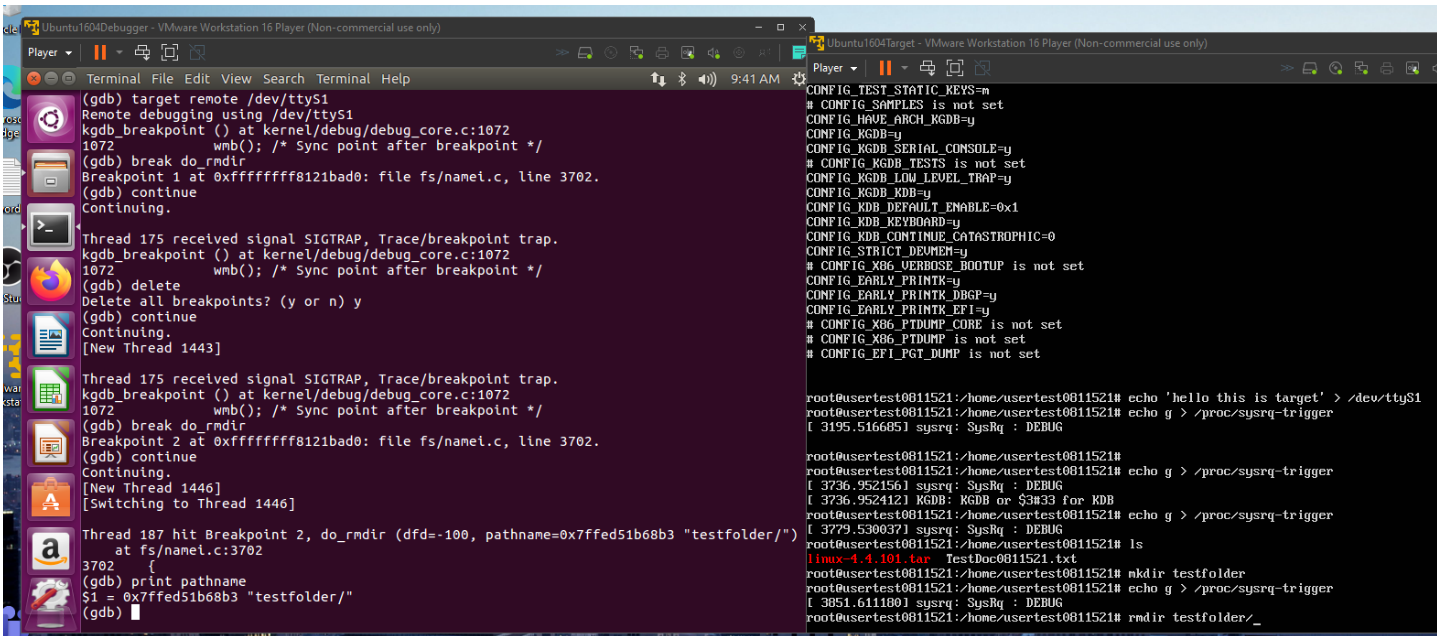
Action: read from a file descriptor

Details:

Screenshot #14

set a breakpoint for **rmdir**

trigger the breakpoint and print the parameter *pathname*

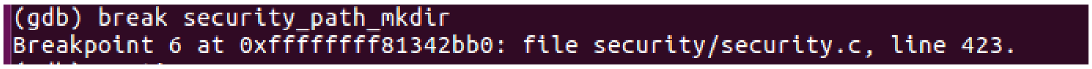

Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player
Terminal File Edit View Search Terminal Help
9:41 AM
Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)

```
(gdb) target remote /dev/ttyS1
Remote debugging using /dev/ttyS1
kgdb_breakpoint () at kernel/debug/debug_core.c:1072
1072            wmb(); /* Sync point after breakpoint */
(gdb) break do_rmdir
Breakpoint 1 at 0xffffffff8121bad0: file fs/namei.c, line 3702.
(gdb) continue
Continuing.

Thread 175 received signal SIGTRAP, Trace/breakpoint trap.
kgdb_breakpoint () at kernel/debug/debug_core.c:1072
1072            wmb(); /* Sync point after breakpoint */
(gdb) delete
Delete all breakpoints? (y or n) y
(gdb) continue
Continuing.
[New Thread 1443]

Thread 175 received signal SIGTRAP, Trace/breakpoint trap.
kgdb_breakpoint () at kernel/debug/debug_core.c:1072
1072            wmb(); /* Sync point after breakpoint */
(gdb) break do_rmdir
Breakpoint 2 at 0xffffffff8121bad0: file fs/namei.c, line 3702.
(gdb) continue
Continuing.
[New Thread 1446]
[Switching to Thread 1446]

Thread 187 hit Breakpoint 2, do_rmdir (dfd=-100, pathname=0x7ffed51b68b3 "testfolder/")
    at fs/namei.c:3702
3702    {
(gdb) print pathname
$1 = 0x7ffed51b68b3 "testfolder/"
(gdb) 
```

```
CONFIG_TEST_STATIC_KEYS=m
# CONFIG_SAMPLES is not set
CONFIG_HAVE_ARCH_KGDB=y
CONFIG_KGDB=y
CONFIG_KGDB_SERIAL_CONSOLE=y
# CONFIG_KGDB_TESTS is not set
CONFIG_KGDB_LOW_LEVEL_TRAP=y
CONFIG_KGDB_KDB=y
CONFIG_KDB_DEFAULT_ENABLE=0x1
CONFIG_KDB_KEYBOARD=y
CONFIG_KDB_CONTINUE_CATASTROPHIC=0
CONFIG_STRICT_DEVMEM=y
# CONFIG_X86_VERBOSE_BOOTUP is not set
CONFIG_EARLY_PRINTK=y
CONFIG_EARLY_PRINTK_DBGP=y
CONFIG_EARLY_PRINTK_EFI=y
# CONFIG_X86_PTDUMP_CORE is not set
# CONFIG_X86_PTDUMP is not set
# CONFIG_EFI_PGT_DUMP is not set


root@usertest0811521:/home/usertest0811521# echo 'hello this is target' > /dev/ttyS1
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[ 3195.516685] sysrq: SysRq : DEBUG

root@usertest0811521:/home/usertest0811521#
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[ 3736.952156] sysrq: SysRq : DEBUG
[ 3736.952412] KGDB: KGDB or $3#33 for KDB
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[ 3779.530037] sysrq: SysRq : DEBUG
root@usertest0811521:/home/usertest0811521# ls
linux-4.4.101.tar  TestDoc0811521.txt
root@usertest0811521:/home/usertest0811521# mkdir testfolder
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[ 3851.611180] sysrq: SysRq : DEBUG
root@usertest0811521:/home/usertest0811521# rmdir testfolder/_
```

Screenshot #15

set one breakpoint for **security_path_mkdir** that will be triggered when executing **mkdir**

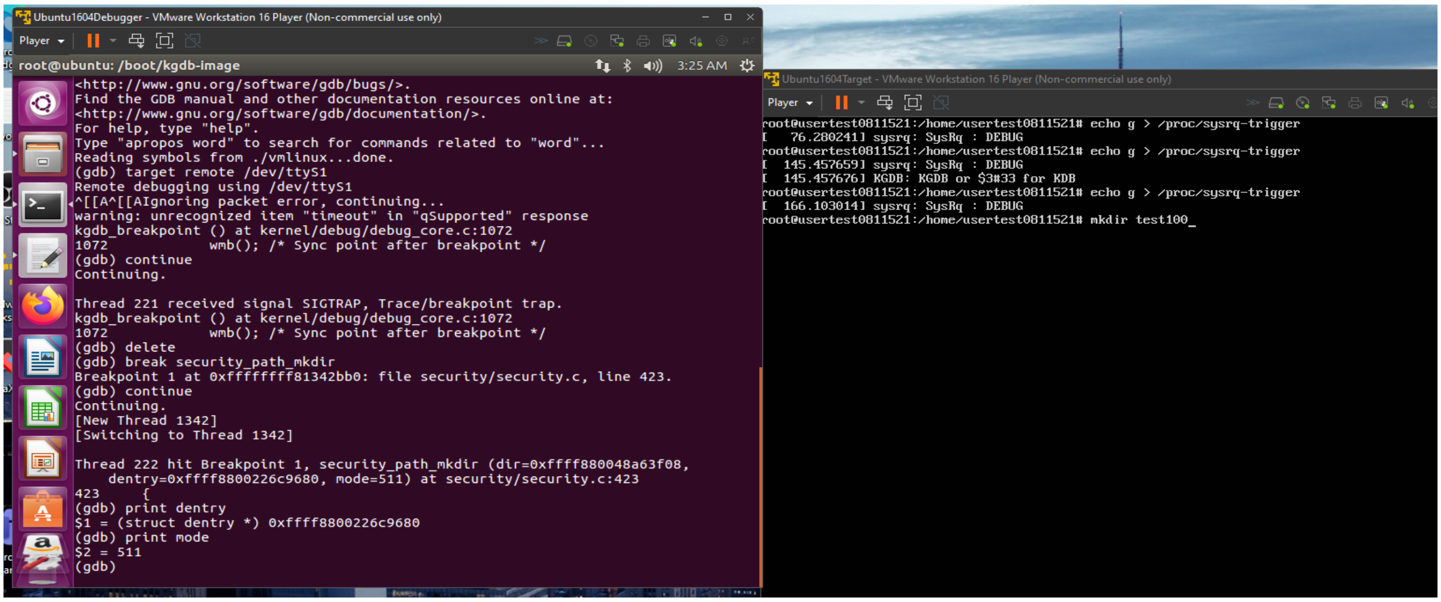


```
(gdb) break security_path_mkdir
Breakpoint 6 at 0xffffffff81342bb0: file security/security.c, line 423.
```

Screenshot #16

parameters of **security_path_mkdir:** path, dentry, and mode

print some parameters of the function to gain a closer look into the information of the breakpoint

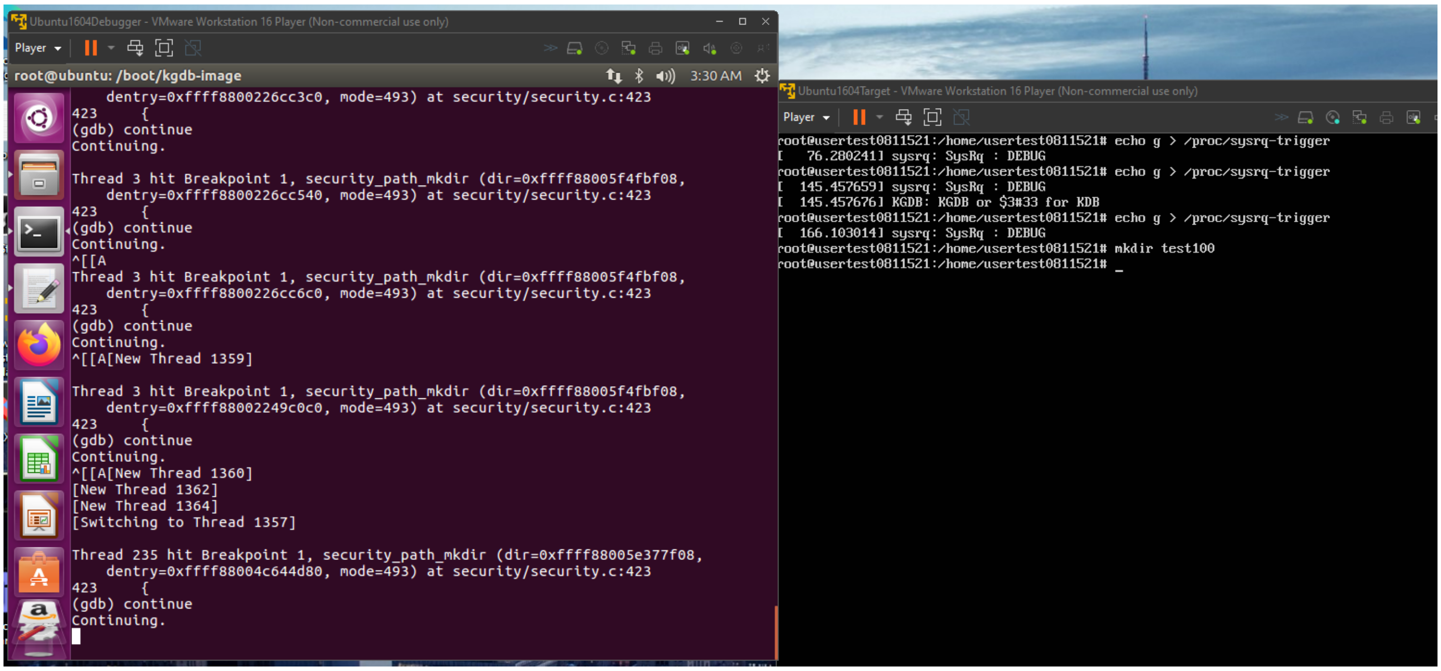

Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player
root@ubuntu: /boot/kgdb-image
3:25 AM
Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)

```
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./vmlinux...done.
(gdb) target remote /dev/ttyS1
Remote debugging using /dev/ttyS1
^[[A^[[AIgnoring packet error, continuing...
warning: unrecognized item "timeout" in "qSupported" response
kgdb_breakpoint () at kernel/debug/debug_core.c:1072
1072            wmb(); /* Sync point after breakpoint */
(gdb) continue
Continuing.

Thread 221 received signal SIGTRAP, Trace/breakpoint trap.
kgdb_breakpoint () at kernel/debug/debug_core.c:1072
1072            wmb(); /* Sync point after breakpoint */
(gdb) delete
(gdb) break security_path_mkdir
Breakpoint 1 at 0xffffffff81342bb0: file security/security.c, line 423.
(gdb) continue
Continuing.
[New Thread 1342]
[Switching to Thread 1342]

Thread 222 hit Breakpoint 1, security_path_mkdir (dir=0xffff880048a63f08,
    dentry=0xffff8800226c9680, mode=511) at security/security.c:423
423     {
(gdb) print dentry
$1 = (struct dentry *) 0xffff8800226c9680
(gdb) print mode
$2 = 511
(gdb)
```

```
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[   76.280241] sysrq: SysRq : DEBUG
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[  145.457659] sysrq: SysRq : DEBUG
[  145.457676] KGDB: KGDB or $3#33 for KDB
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[  166.103014] sysrq: SysRq : DEBUG
root@usertest0811521:/home/usertest0811521# mkdir test100_
```

Screenshot #17

after several **continue**, the target machine regains control.



Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player
root@ubuntu: /boot/kgdb-image
3:30 AM

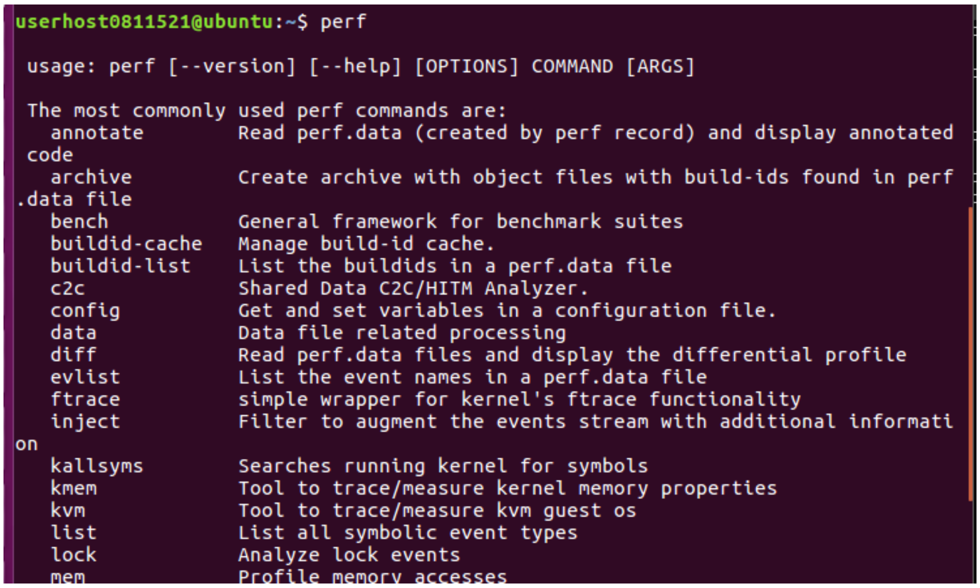
```
    dentry=0xffff8800226cc3c0, mode=493) at security/security.c:423
423     {
(gdb) continue
Continuing.

Thread 3 hit Breakpoint 1, security_path_mkdir (dir=0xffff88005f4fbf08,
    dentry=0xffff8800226cc540, mode=493) at security/security.c:423
423     {
(gdb) continue
Continuing.
^[[A
Thread 3 hit Breakpoint 1, security_path_mkdir (dir=0xffff88005f4fbf08,
    dentry=0xffff8800226cc6c0, mode=493) at security/security.c:423
423     {
(gdb) continue
Continuing.
^[[A[New Thread 1359]

Thread 3 hit Breakpoint 1, security_path_mkdir (dir=0xffff88005f4fbf08,
    dentry=0xffff88002249c0c0, mode=493) at security/security.c:423
423     {
(gdb) continue
Continuing.
^[[A[New Thread 1360]
[New Thread 1362]
[New Thread 1364]
[Switching to Thread 1357]

Thread 235 hit Breakpoint 1, security_path_mkdir (dir=0xffff88005e377f08,
    dentry=0xffff88004c644d80, mode=493) at security/security.c:423
423     {
(gdb) continue
Continuing.
```

Ubuntu1604Target - VMware Workstation 16 Player (Non-commercial use only)
Player

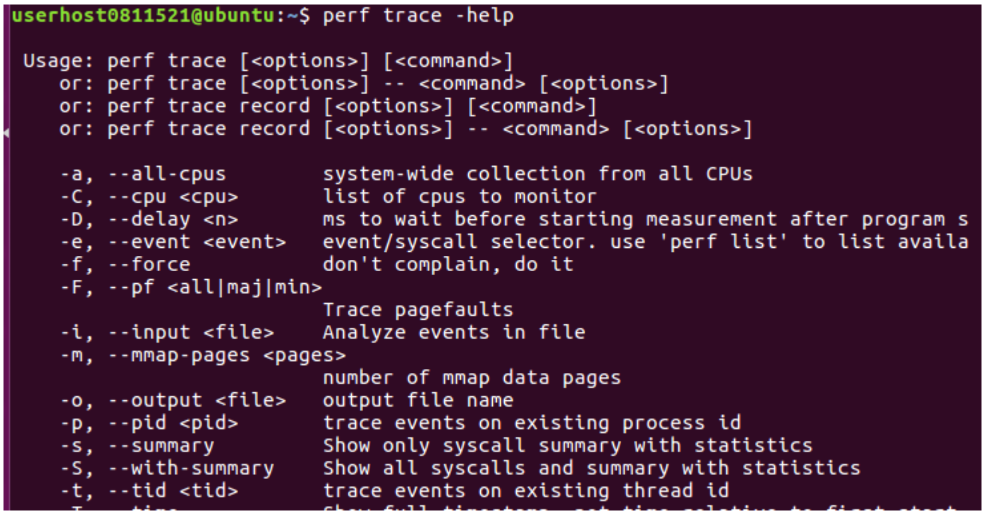
```
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[   76.280241] sysrq: SysRq : DEBUG
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[  145.457659] sysrq: SysRq : DEBUG
[  145.457676] KGDB: KGDB or $3#33 for KDB
root@usertest0811521:/home/usertest0811521# echo g > /proc/sysrq-trigger
[  166.103014] sysrq: SysRq : DEBUG
root@usertest0811521:/home/usertest0811521# mkdir test100
root@usertest0811521:/home/usertest0811521# _
```

Screenshot #18

the perf command



```
userhost0811521@ubuntu:~$ perf

 usage: perf [--version] [--help] [OPTIONS] COMMAND [ARGS]

 The most commonly used perf commands are:
   annotate        Read perf.data (created by perf record) and display annotated
 code
   archive         Create archive with object files with build-ids found in perf
.data file
   bench           General framework for benchmark suites
   buildid-cache   Manage build-id cache.
   buildid-list    List the buildids in a perf.data file
   c2c             Shared Data C2C/HITM Analyzer.
   config          Get and set variables in a configuration file.
   data            Data file related processing
   diff            Read perf.data files and display the differential profile
   evlist          List the event names in a perf.data file
   ftrace          simple wrapper for kernel's ftrace functionality
   inject          Filter to augment the events stream with additional informati
on
   kallsyms        Searches running kernel for symbols
   kmem            Tool to trace/measure kernel memory properties
   kvm             Tool to trace/measure kvm guest os
   list            List all symbolic event types
   lock            Analyze lock events
   mem             Profile memory accesses
```

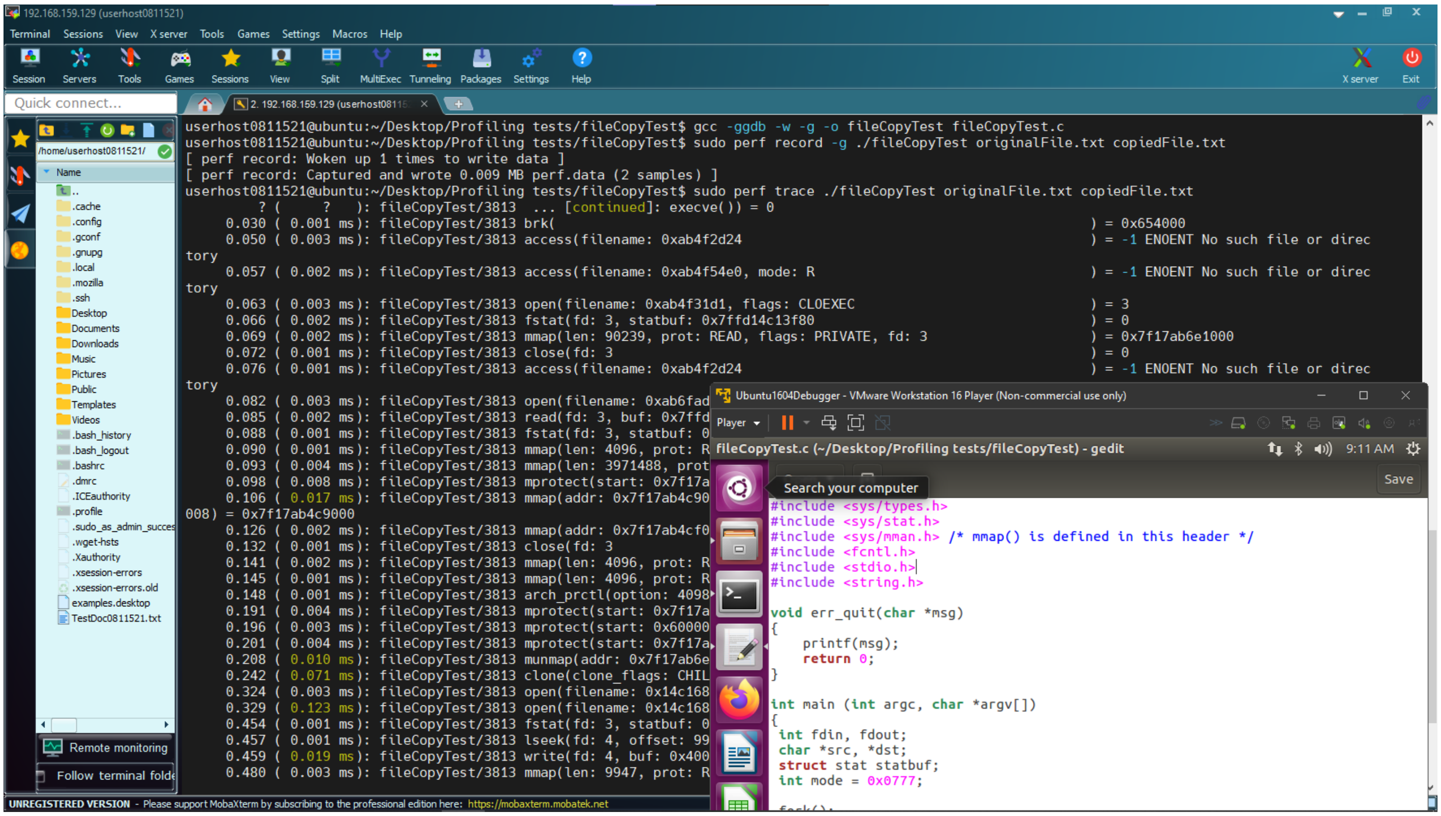


```
userhost0811521@ubuntu:~$ perf trace -help

 Usage: perf trace [<options>] [<command>]
    or: perf trace [<options>] -- <command> [<options>]
    or: perf trace record [<options>] [<command>]
    or: perf trace record [<options>] -- <command> [<options>]

    -a, --all-cpus        system-wide collection from all CPUs
    -C, --cpu <cpu>       list of cpus to monitor
    -D, --delay <n>       ms to wait before starting measurement after program s
    -e, --event <event>   event/syscall selector. use 'perf list' to list availa
    -f, --force           don't complain, do it
    -F, --pf <all|maj|min>
                          Trace pagefaults
    -i, --input <file>    Analyze events in file
    -m, --mmap-pages <pages>
                          number of mmap data pages
    -o, --output <file>   output file name
    -p, --pid <pid>       trace events on existing process id
    -s, --summary         Show only syscall summary with statistics
    -S, --with-summary    Show all syscalls and summary with statistics
    -t, --tid <tid>       trace events on existing thread id
    -T, --time            Show full timestamp, not time relative to first start
```

Screenshot #19

This is the execution of **fileCopyTest.c**.

the execution result of perf, the trace result, and the generated files

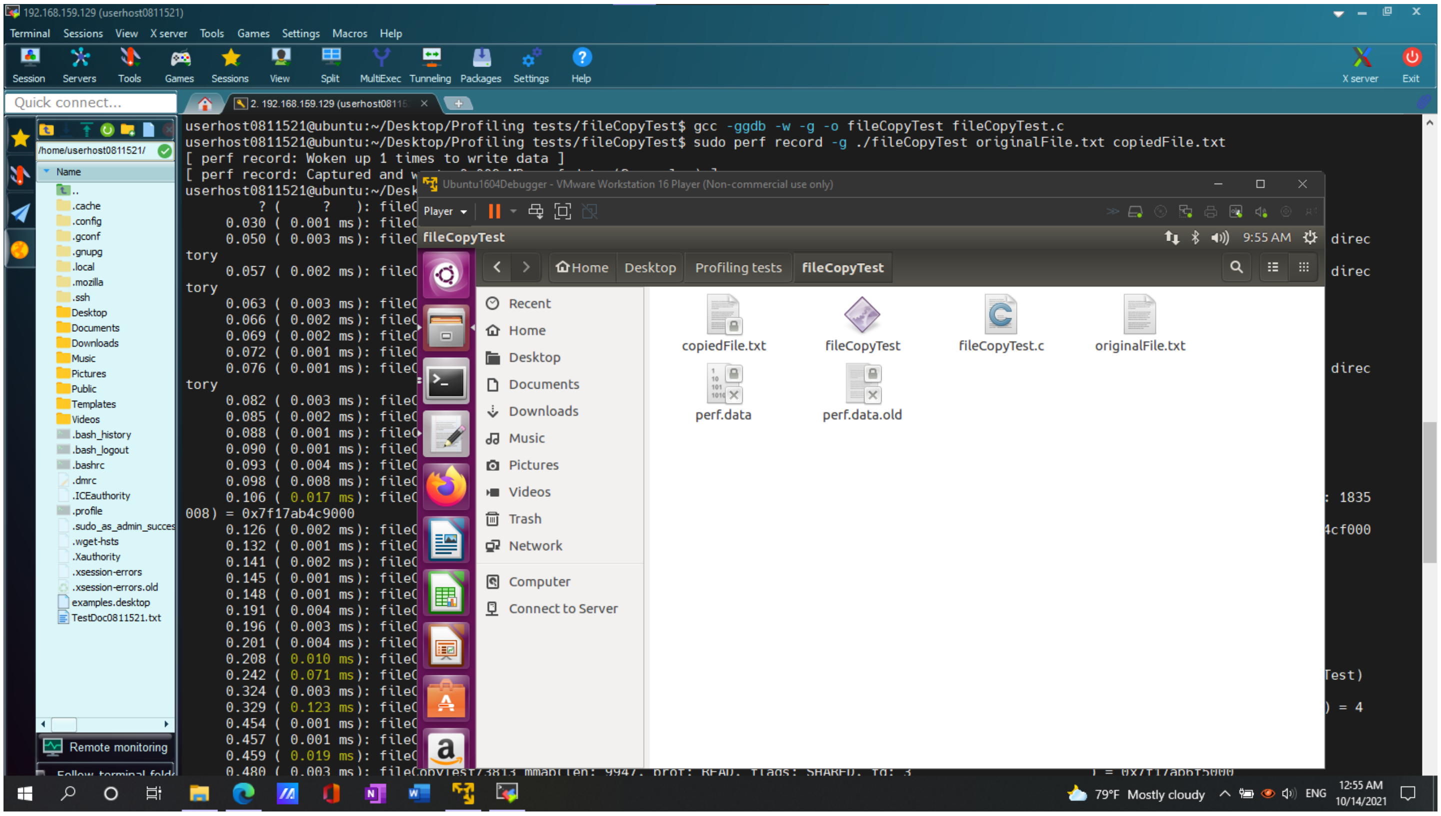


```
userhost0811521@ubuntu:~/Desktop/Profiling tests/fileCopyTest$ gcc -ggdb -w -g -o fileCopyTest fileCopyTest.c
userhost0811521@ubuntu:~/Desktop/Profiling tests/fileCopyTest$ sudo perf record -g ./fileCopyTest originalFile.txt copiedFile.txt
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.009 MB perf.data (2 samples) ]
userhost0811521@ubuntu:~/Desktop/Profiling tests/fileCopyTest$ sudo perf trace ./fileCopyTest originalFile.txt copiedFile.txt
         ? (     ?   ): fileCopyTest/3813  ... [continued]: execve()) = 0
     0.030 ( 0.001 ms): fileCopyTest/3813 brk(                                                    ) = 0x654000
     0.050 ( 0.003 ms): fileCopyTest/3813 access(filename: 0xab4f2d24                             ) = -1 ENOENT No such file or direc
tory
     0.057 ( 0.002 ms): fileCopyTest/3813 access(filename: 0xab4f54e0, mode: R                    ) = -1 ENOENT No such file or direc
tory
     0.063 ( 0.003 ms): fileCopyTest/3813 open(filename: 0xab4f31d1, flags: CLOEXEC               ) = 3
     0.066 ( 0.002 ms): fileCopyTest/3813 fstat(fd: 3, statbuf: 0x7ffd14c13f80                    ) = 0
     0.069 ( 0.002 ms): fileCopyTest/3813 mmap(len: 90239, prot: READ, flags: PRIVATE, fd: 3      ) = 0x7f17ab6e1000
     0.072 ( 0.001 ms): fileCopyTest/3813 close(fd: 3                                             ) = 0
     0.076 ( 0.001 ms): fileCopyTest/3813 access(filename: 0xab4f2d24                             ) = -1 ENOENT No such file or direc
tory
     0.082 ( 0.003 ms): fileCopyTest/3813 open(filename: 0xab6fad
     0.085 ( 0.002 ms): fileCopyTest/3813 read(fd: 3, buf: 0x7ffd
     0.088 ( 0.001 ms): fileCopyTest/3813 fstat(fd: 3, statbuf: 0
     0.090 ( 0.001 ms): fileCopyTest/3813 mmap(len: 4096, prot: R
     0.093 ( 0.004 ms): fileCopyTest/3813 mmap(len: 3971488, prot
     0.098 ( 0.008 ms): fileCopyTest/3813 mprotect(start: 0x7f17a
     0.106 ( 0.017 ms): fileCopyTest/3813 mmap(addr: 0x7f17ab4c90
008) = 0x7f17ab4c9000
     0.126 ( 0.002 ms): fileCopyTest/3813 mmap(addr: 0x7f17ab4cf0
     0.132 ( 0.001 ms): fileCopyTest/3813 close(fd: 3
     0.141 ( 0.002 ms): fileCopyTest/3813 mmap(len: 4096, prot: R
     0.145 ( 0.001 ms): fileCopyTest/3813 mmap(len: 4096, prot: R
     0.148 ( 0.001 ms): fileCopyTest/3813 arch_prctl(option: 4098
     0.191 ( 0.004 ms): fileCopyTest/3813 mprotect(start: 0x7f17a
     0.196 ( 0.003 ms): fileCopyTest/3813 mprotect(start: 0x60000
     0.201 ( 0.004 ms): fileCopyTest/3813 mprotect(start: 0x7f17a
     0.208 ( 0.010 ms): fileCopyTest/3813 munmap(addr: 0x7f17ab6e
     0.242 ( 0.071 ms): fileCopyTest/3813 clone(clone_flags: CHIL
     0.324 ( 0.003 ms): fileCopyTest/3813 open(filename: 0x14c168
     0.329 ( 0.123 ms): fileCopyTest/3813 open(filename: 0x14c168
     0.454 ( 0.001 ms): fileCopyTest/3813 fstat(fd: 3, statbuf: 0
     0.457 ( 0.001 ms): fileCopyTest/3813 lseek(fd: 4, offset: 99
     0.459 ( 0.019 ms): fileCopyTest/3813 write(fd: 4, buf: 0x400
     0.480 ( 0.003 ms): fileCopyTest/3813 mmap(len: 9947, prot: R
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

fileCopyTest.c (~/Desktop/Profiling tests/fileCopyTest) - gedit

```
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/mman.h> /* mmap() is defined in this header */
#include <fcntl.h>
#include <stdio.h>
#include <string.h>

void err_quit(char *msg)
{
    printf(msg);
    return 0;
}

int main (int argc, char *argv[])
{
 int fdin, fdout;
 char *src, *dst;
 struct stat statbuf;
 int mode = 0x0777;
```



```
userhost0811521@ubuntu:~/Desktop/Profiling tests/fileCopyTest$ gcc -ggdb -w -g -o fileCopyTest fileCopyTest.c
userhost0811521@ubuntu:~/Desktop/Profiling tests/fileCopyTest$ sudo perf record -g ./fileCopyTest originalFile.txt copiedFile.txt
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and w
userhost0811521@ubuntu:~/Desk
     0.480 ( 0.003 ms): fileCopyTest/3813 mmap(len: 9947, prot: READ, flags: SHARED, fd: 3          ) = 0x7f17ab6f5000
```

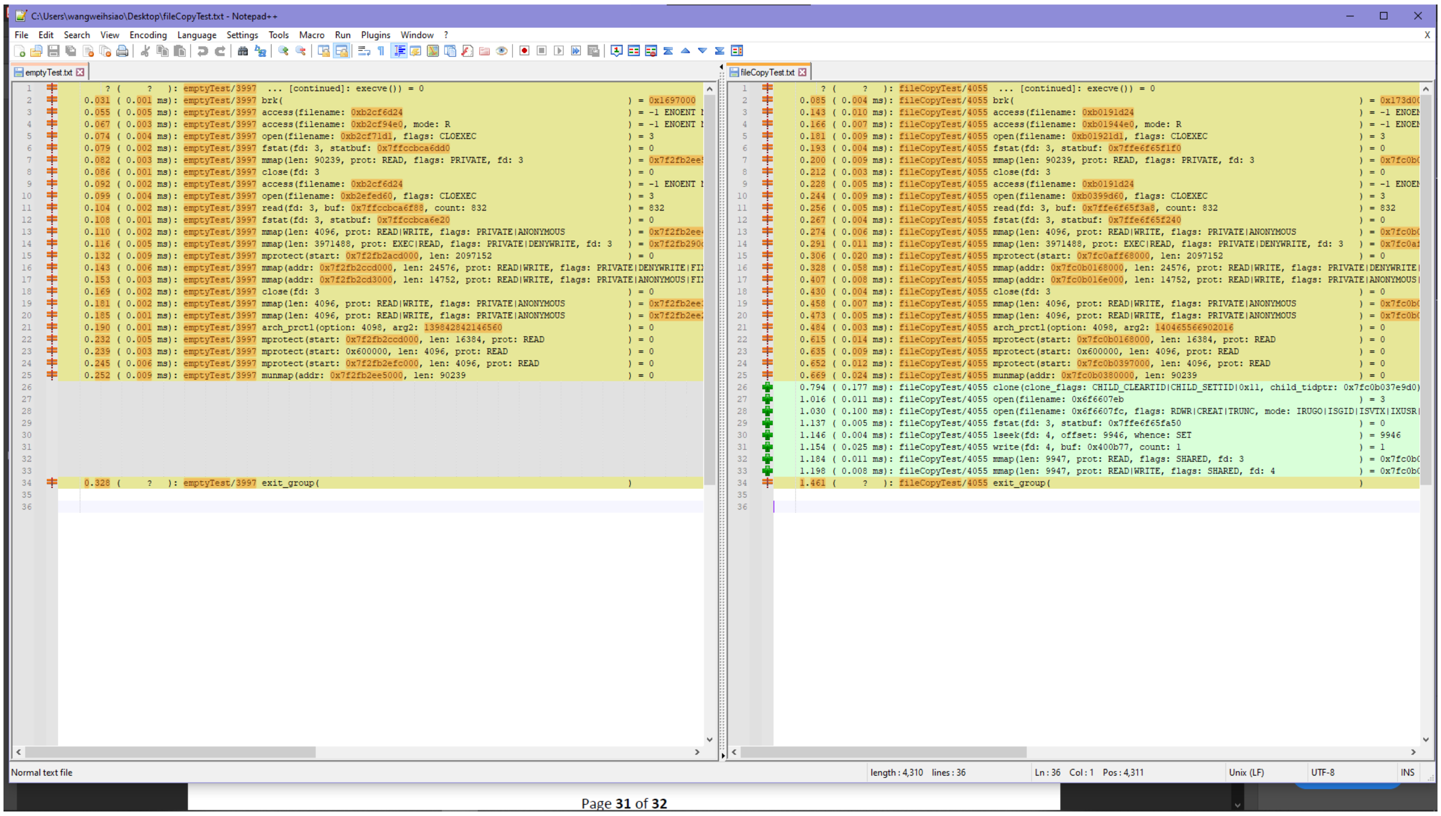
fileCopyTest

copiedFile.txt fileCopyTest fileCopyTest.c originalFile.txt perf.data perf.data.old

<span style="color:red">Screenshot #20</span>

compare the trace results of **emptyTest.c** and **fileCopyTest.c**

Since **fileCopyTest.c** executes a few more commands such as calling **fork()**, so its trace result shows additional events.



C:\Users\wangweihsiao\Desktop\fileCopyTest.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
emptyTest.txt
fileCopyTest.txt

```
      ? (     ?   ): emptyTest/3997  ... [continued]: execve()) = 0
  0.031 ( 0.001 ms): emptyTest/3997 brk(                                                      ) = 0x1697000
  0.055 ( 0.005 ms): emptyTest/3997 access(filename: 0xb2cf6d24                               ) = -1 ENOENT
  0.067 ( 0.003 ms): emptyTest/3997 access(filename: 0xb2cf94e0, mode: R                      ) = -1 ENOENT
  0.074 ( 0.004 ms): emptyTest/3997 open(filename: 0xb2cf71d1, flags: CLOEXEC                 ) = 3
  0.079 ( 0.002 ms): emptyTest/3997 fstat(fd: 3, statbuf: 0x7ffccbca6dd0                      ) = 0
  0.082 ( 0.003 ms): emptyTest/3997 mmap(len: 90239, prot: READ, flags: PRIVATE, fd: 3        ) = 0x7f2fb2ee5
  0.086 ( 0.001 ms): emptyTest/3997 close(fd: 3                                               ) = 0
  0.092 ( 0.002 ms): emptyTest/3997 access(filename: 0xb2cf6d24                               ) = -1 ENOENT
  0.099 ( 0.004 ms): emptyTest/3997 open(filename: 0xb2efed60, flags: CLOEXEC                 ) = 3
  0.104 ( 0.002 ms): emptyTest/3997 read(fd: 3, buf: 0x7ffccbca6f88, count: 832               ) = 832
  0.108 ( 0.001 ms): emptyTest/3997 fstat(fd: 3, statbuf: 0x7ffccbca6e20                      ) = 0
  0.110 ( 0.002 ms): emptyTest/3997 mmap(len: 4096, prot: READ|WRITE, flags: PRIVATE|ANONYMOUS ) = 0x7f2fb2ee
  0.116 ( 0.005 ms): emptyTest/3997 mmap(len: 3971488, prot: EXEC|READ, flags: PRIVATE|DENYWRITE, fd: 3 ) = 0x7f2fb290
  0.132 ( 0.009 ms): emptyTest/3997 mprotect(start: 0x7f2fb2acd000, len: 2097152             ) = 0
  0.143 ( 0.006 ms): emptyTest/3997 mmap(addr: 0x7f2fb2ccd000, len: 24576, prot: READ|WRITE, flags: PRIVATE|DENYWRITE|FI
  0.153 ( 0.003 ms): emptyTest/3997 mmap(addr: 0x7f2fb2cd3000, len: 14752, prot: READ|WRITE, flags: PRIVATE|ANONYMOUS|FI
  0.169 ( 0.002 ms): emptyTest/3997 close(fd: 3                                               ) = 0
  0.181 ( 0.002 ms): emptyTest/3997 mmap(len: 4096, prot: READ|WRITE, flags: PRIVATE|ANONYMOUS ) = 0x7f2fb2ee
  0.185 ( 0.001 ms): emptyTest/3997 mmap(len: 4096, prot: READ|WRITE, flags: PRIVATE|ANONYMOUS ) = 0x7f2fb2ee
  0.190 ( 0.001 ms): emptyTest/3997 arch_prctl(option: 4098, arg2: 139842842146560           ) = 0
  0.232 ( 0.005 ms): emptyTest/3997 mprotect(start: 0x7f2fb2ccd000, len: 16384, prot: READ   ) = 0
  0.239 ( 0.003 ms): emptyTest/3997 mprotect(start: 0x600000, len: 4096, prot: READ          ) = 0
  0.245 ( 0.006 ms): emptyTest/3997 mprotect(start: 0x7f2fb2efc000, len: 4096, prot: READ    ) = 0
  0.252 ( 0.009 ms): emptyTest/3997 munmap(addr: 0x7f2fb2ee5000, len: 90239                  ) = 0








  0.328 (     ?   ): emptyTest/3997 exit_group(                                               )
```

```
      ? (     ?   ): fileCopyTest/4055  ... [continued]: execve()) = 0
  0.085 ( 0.004 ms): fileCopyTest/4055 brk(                                                      ) = 0x173d0
  0.143 ( 0.010 ms): fileCopyTest/4055 access(filename: 0xb0191d24                               ) = -1 ENOEN
  0.166 ( 0.007 ms): fileCopyTest/4055 access(filename: 0xb01944e0, mode: R                      ) = -1 ENOEN
  0.181 ( 0.009 ms): fileCopyTest/4055 open(filename: 0xb01921d1, flags: CLOEXEC                 ) = 3
  0.193 ( 0.004 ms): fileCopyTest/4055 fstat(fd: 3, statbuf: 0x7ffe6f65f1f0                      ) = 0
  0.200 ( 0.009 ms): fileCopyTest/4055 mmap(len: 90239, prot: READ, flags: PRIVATE, fd: 3        ) = 0x7fc0b0
  0.212 ( 0.003 ms): fileCopyTest/4055 close(fd: 3                                               ) = 0
  0.228 ( 0.005 ms): fileCopyTest/4055 access(filename: 0xb0191d24                               ) = -1 ENOEN
  0.244 ( 0.009 ms): fileCopyTest/4055 open(filename: 0xb0399d60, flags: CLOEXEC                 ) = 3
  0.256 ( 0.005 ms): fileCopyTest/4055 read(fd: 3, buf: 0x7ffe6f65f3a8, count: 832               ) = 832
  0.267 ( 0.004 ms): fileCopyTest/4055 fstat(fd: 3, statbuf: 0x7ffe6f65f240                      ) = 0
  0.274 ( 0.006 ms): fileCopyTest/4055 mmap(len: 4096, prot: READ|WRITE, flags: PRIVATE|ANONYMOUS ) = 0x7fc0b0
  0.291 ( 0.011 ms): fileCopyTest/4055 mmap(len: 3971488, prot: EXEC|READ, flags: PRIVATE|DENYWRITE, fd: 3 ) = 0x7fc0a
  0.306 ( 0.020 ms): fileCopyTest/4055 mprotect(start: 0x7fc0aff68000, len: 2097152             ) = 0
  0.328 ( 0.058 ms): fileCopyTest/4055 mmap(addr: 0x7fc0b0168000, len: 24576, prot: READ|WRITE, flags: PRIVATE|DENYWRITE|
  0.407 ( 0.008 ms): fileCopyTest/4055 mmap(addr: 0x7fc0b016e000, len: 14752, prot: READ|WRITE, flags: PRIVATE|ANONYMOUS|
  0.430 ( 0.004 ms): fileCopyTest/4055 close(fd: 3                                               ) = 0
  0.458 ( 0.007 ms): fileCopyTest/4055 mmap(len: 4096, prot: READ|WRITE, flags: PRIVATE|ANONYMOUS ) = 0x7fc0b0
  0.473 ( 0.005 ms): fileCopyTest/4055 mmap(len: 4096, prot: READ|WRITE, flags: PRIVATE|ANONYMOUS ) = 0x7fc0b0
  0.484 ( 0.003 ms): fileCopyTest/4055 arch_prctl(option: 4098, arg2: 140465566902016           ) = 0
  0.615 ( 0.014 ms): fileCopyTest/4055 mprotect(start: 0x7fc0b0168000, len: 16384, prot: READ   ) = 0
  0.635 ( 0.009 ms): fileCopyTest/4055 mprotect(start: 0x600000, len: 4096, prot: READ          ) = 0
  0.652 ( 0.012 ms): fileCopyTest/4055 mprotect(start: 0x7fc0b0397000, len: 4096, prot: READ    ) = 0
  0.669 ( 0.024 ms): fileCopyTest/4055 munmap(addr: 0x7fc0b0380000, len: 90239                  ) = 0
  0.794 ( 0.177 ms): fileCopyTest/4055 clone(clone_flags: CHILD_CLEARTID|CHILD_SETTID|0x11, child_tidptr: 0x7fc0b037e9d0)
  1.016 ( 0.011 ms): fileCopyTest/4055 open(filename: 0x6f6607eb                                 ) = 3
  1.030 ( 0.100 ms): fileCopyTest/4055 open(filename: 0x6f6607fc, flags: RDWR|CREAT|TRUNC, mode: IRUGO|ISGID|ISVTX|IXUSR|
  1.137 ( 0.005 ms): fileCopyTest/4055 fstat(fd: 3, statbuf: 0x7ffe6f65fa50                      ) = 0
  1.146 ( 0.004 ms): fileCopyTest/4055 lseek(fd: 4, offset: 9946, whence: SET                    ) = 9946
  1.154 ( 0.025 ms): fileCopyTest/4055 write(fd: 4, buf: 0x400b77, count: 1                      ) = 1
  1.184 ( 0.011 ms): fileCopyTest/4055 mmap(len: 9947, prot: READ, flags: SHARED, fd: 3          ) = 0x7fc0b0
  1.198 ( 0.008 ms): fileCopyTest/4055 mmap(len: 9947, prot: READ|WRITE, flags: SHARED, fd: 4    ) = 0x7fc0b0
  1.461 (     ?   ): fileCopyTest/4055 exit_group(                                               )
```

(p.32)

a. Will the functions' execution time be longer if the file is bigger?

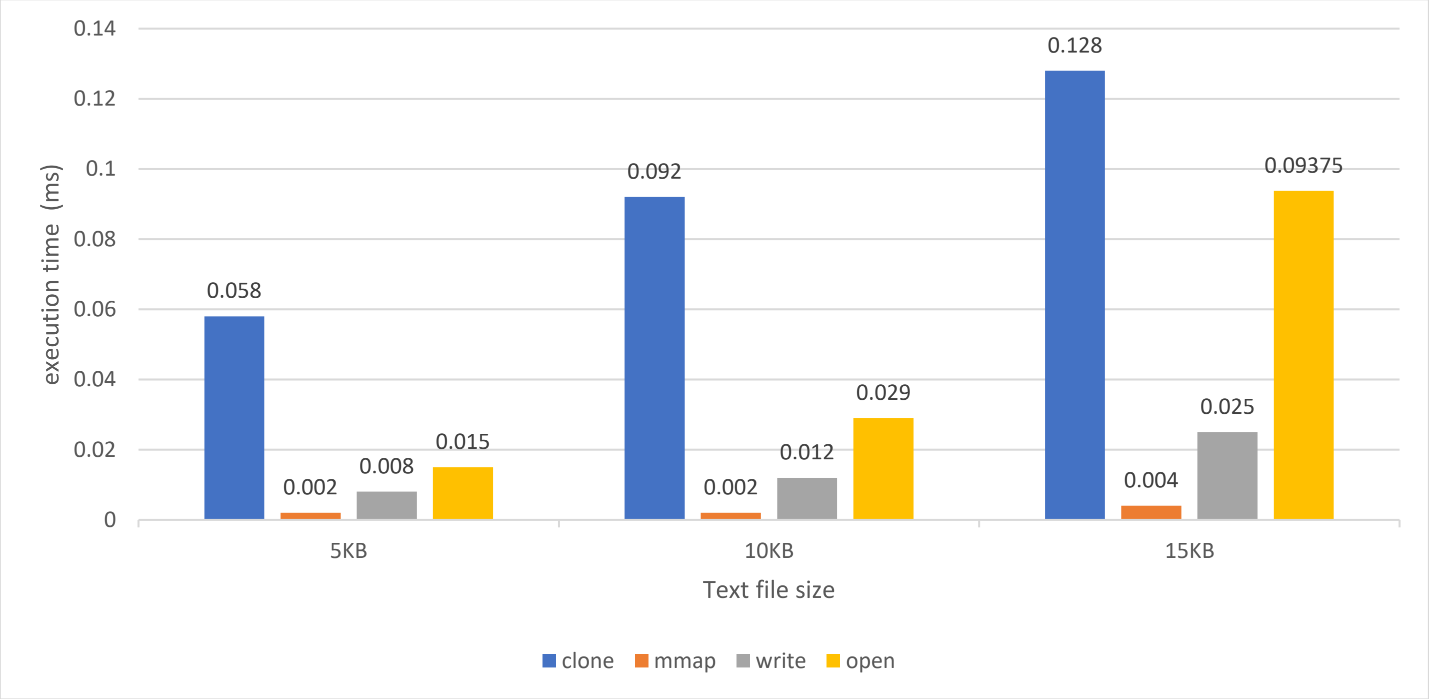
Ans: According to my experiment, the executions of **clone**, **open**, and **write** take longer time when the text file is larger. The executions of **mmap** for different file size take roughly the same amount of time. But there is some time when the executions of same function take same amount of time for different file sizes; I guess the file sizes are not large enough to generate distinct differences.

b. How is the behavior of each function? Sort them from slowest to fastest.

Ans: from fastest to slowest**: mmap, write, open, clone**

c. Create a graph of file size (in bytes) vs. execution time (ms) of these four functions, using 3 different file sizes.

Screenshot #21



d. Perf also has the report command:

i. What is it for?

Ans: Show details of performance of processes and overhead of each function

ii. For fileCopyTest, show and interpret the results.

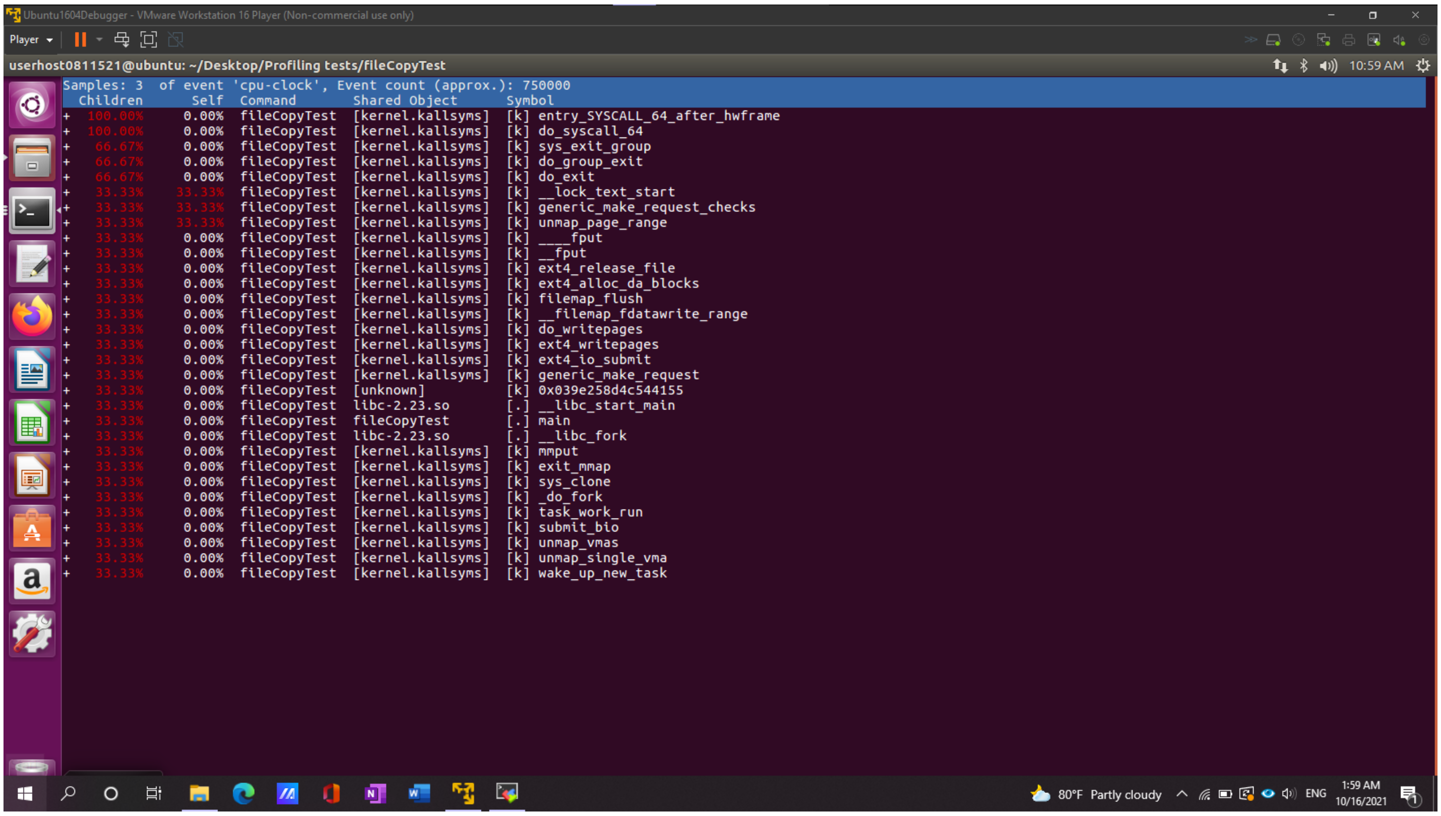
Ans: The following functions have the highest overhead:

__lock_text_start

generic_make_request_checks

unmap_page_range

Screenshot #22 [perf report]


Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player
10:59 AM
80°F Partly cloudy
ENG
1:59 AM
10/16/2021

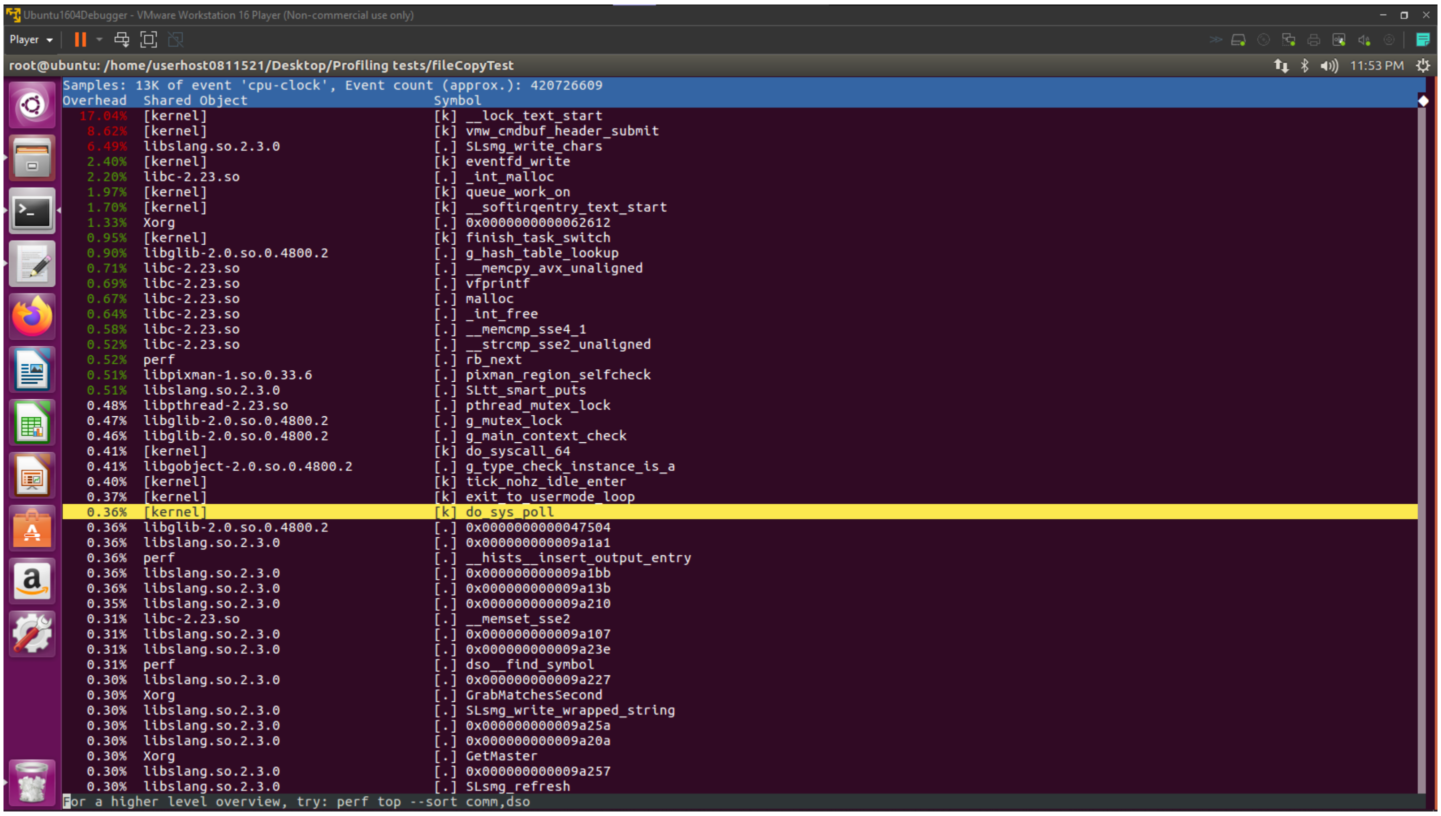
```
userhost0811521@ubuntu: ~/Desktop/Profiling tests/fileCopyTest
Samples: 3  of event 'cpu-clock', Event count (approx.): 750000
  Children      Self  Command       Shared Object       Symbol
+  100.00%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] entry_SYSCALL_64_after_hwframe
+  100.00%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] do_syscall_64
+   66.67%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] sys_exit_group
+   66.67%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] do_group_exit
+   66.67%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] do_exit
+   33.33%    33.33%  fileCopyTest  [kernel.kallsyms]  [k] __lock_text_start
+   33.33%    33.33%  fileCopyTest  [kernel.kallsyms]  [k] generic_make_request_checks
+   33.33%    33.33%  fileCopyTest  [kernel.kallsyms]  [k] unmap_page_range
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] ____fput
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] __fput
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] ext4_release_file
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] ext4_alloc_da_blocks
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] filemap_flush
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] __filemap_fdatawrite_range
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] do_writepages
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] ext4_writepages
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] ext4_io_submit
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] generic_make_request
+   33.33%     0.00%  fileCopyTest  [unknown]          [k] 0x039e258d4c544155
+   33.33%     0.00%  fileCopyTest  libc-2.23.so       [.] __libc_start_main
+   33.33%     0.00%  fileCopyTest  fileCopyTest       [.] main
+   33.33%     0.00%  fileCopyTest  libc-2.23.so       [.] __libc_fork
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] mmput
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] exit_mmap
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] sys_clone
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] _do_fork
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] task_work_run
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] submit_bio
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] unmap_vmas
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] unmap_single_vma
+   33.33%     0.00%  fileCopyTest  [kernel.kallsyms]  [k] wake_up_new_task
```

e. Perf has more commands (Section 5.1 step 4). Select another command (besides report, trace and record), explain what it is for and show how to use it. Create your own scenario.

Ans: **perf top**

function: real-time analysis of performance

Screenshot #23 [perf top]

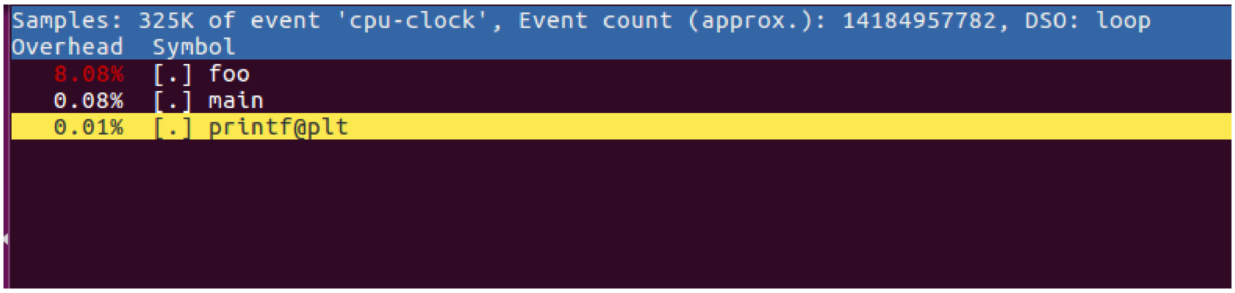


Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player
root@ubuntu: /home/userhost0811521/Desktop/Profiling tests/fileCopyTest
11:53 PM

```
Samples: 13K of event 'cpu-clock', Event count (approx.): 420726609
Overhead  Shared Object                               Symbol
  17.04%  [kernel]                                    [k] __lock_text_start
   8.62%  [kernel]                                    [k] vmw_cmdbuf_header_submit
   6.49%  libslang.so.2.3.0                           [.] SLsmg_write_chars
   2.40%  [kernel]                                    [k] eventfd_write
   2.20%  libc-2.23.so                                [.] _int_malloc
   1.97%  [kernel]                                    [k] queue_work_on
   1.70%  [kernel]                                    [k] __softirqentry_text_start
   1.33%  Xorg                                        [.] 0x0000000000062612
   0.95%  [kernel]                                    [k] finish_task_switch
   0.90%  libglib-2.0.so.0.4800.2                     [.] g_hash_table_lookup
   0.71%  libc-2.23.so                                [.] __memcpy_avx_unaligned
   0.69%  libc-2.23.so                                [.] vfprintf
   0.67%  libc-2.23.so                                [.] malloc
   0.64%  libc-2.23.so                                [.] _int_free
   0.58%  libc-2.23.so                                [.] __memcmp_sse4_1
   0.52%  libc-2.23.so                                [.] __strcmp_sse2_unaligned
   0.52%  perf                                        [.] rb_next
   0.51%  libpixman-1.so.0.33.6                       [.] pixman_region_selfcheck
   0.51%  libslang.so.2.3.0                           [.] SLtt_smart_puts
   0.48%  libpthread-2.23.so                          [.] pthread_mutex_lock
   0.47%  libglib-2.0.so.0.4800.2                     [.] g_mutex_lock
   0.46%  libglib-2.0.so.0.4800.2                     [.] g_main_context_check
   0.41%  [kernel]                                    [k] do_syscall_64
   0.41%  libgobject-2.0.so.0.4800.2                  [.] g_type_check_instance_is_a
   0.40%  [kernel]                                    [k] tick_nohz_idle_enter
   0.37%  [kernel]                                    [k] exit_to_usermode_loop
   0.36%  [kernel]                                    [k] do_sys_poll
   0.36%  libglib-2.0.so.0.4800.2                     [.] 0x0000000000047504
   0.36%  libslang.so.2.3.0                           [.] 0x000000000009a1a1
   0.36%  perf                                        [.] __hists__insert_output_entry
   0.36%  libslang.so.2.3.0                           [.] 0x000000000009a1bb
   0.36%  libslang.so.2.3.0                           [.] 0x000000000009a13b
   0.35%  libslang.so.2.3.0                           [.] 0x000000000009a210
   0.31%  libc-2.23.so                                [.] __memset_sse2
   0.31%  libslang.so.2.3.0                           [.] 0x000000000009a107
   0.31%  libslang.so.2.3.0                           [.] 0x000000000009a23e
   0.31%  perf                                        [.] dso__find_symbol
   0.30%  libslang.so.2.3.0                           [.] 0x000000000009a227
   0.30%  Xorg                                        [.] GrabMatchesSecond
   0.30%  libslang.so.2.3.0                           [.] SLsmg_write_wrapped_string
   0.30%  libslang.so.2.3.0                           [.] 0x000000000009a25a
   0.30%  libslang.so.2.3.0                           [.] 0x000000000009a20a
   0.30%  Xorg                                        [.] GetMaster
   0.30%  libslang.so.2.3.0                           [.] 0x000000000009a257
   0.30%  libslang.so.2.3.0                           [.] SLsmg_refresh
For a higher level overview, try: perf top --sort comm,dso
```

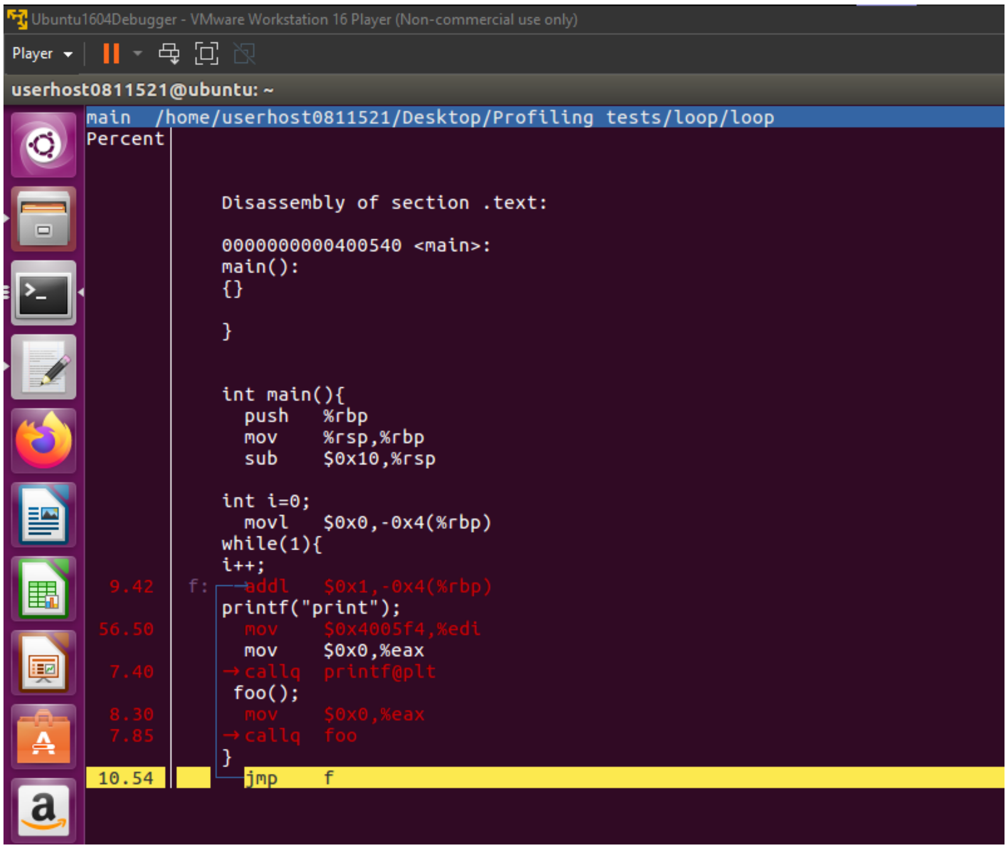
Screenshot #24 [zoom into DSO loop]

I execute a file with infinite while loops that keep calling a function foo().



```
Samples: 325K of event 'cpu-clock', Event count (approx.): 14184957782, DSO: loop
Overhead  Symbol
   8.08%  [.] foo
   0.08%  [.] main
   0.01%  [.] printf@plt
```

(Below)In the main function, we can inspect the parts that occupy the resources.


Ubuntu1604Debugger - VMware Workstation 16 Player (Non-commercial use only)
Player

```
userhost0811521@ubuntu: ~
main  /home/userhost0811521/Desktop/Profiling tests/loop/loop
Percent│
       │
       │
       │     Disassembly of section .text:
       │
       │     0000000000400540 <main>:
       │     main():
       │     {}
       │
       │     }
       │
       │
       │     int main(){
       │       push   %rbp
       │       mov    %rsp,%rbp
       │       sub    $0x10,%rsp
       │
       │     int i=0;
       │       movl   $0x0,-0x4(%rbp)
       │     while(1){
       │     i++;
  9.42 │ f:┌─→addl   $0x1,-0x4(%rbp)
       │   │ printf("print");
 56.50 │   │  mov    $0x4005f4,%edi
       │   │  mov    $0x0,%eax
  7.40 │   │→ callq  printf@plt
       │   │  foo();
  8.30 │   │  mov    $0x0,%eax
  7.85 │   │→ callq  foo
       │   │ }
 10.54 │   └──jmp    f
```

Sources:

[1]: https://www.trendmicro.com/en_us/research/17/a/practical-android-debugging-via-kgdb.html