How to add your system call the Linux OS kernel?

اميرة يحيى سليمان

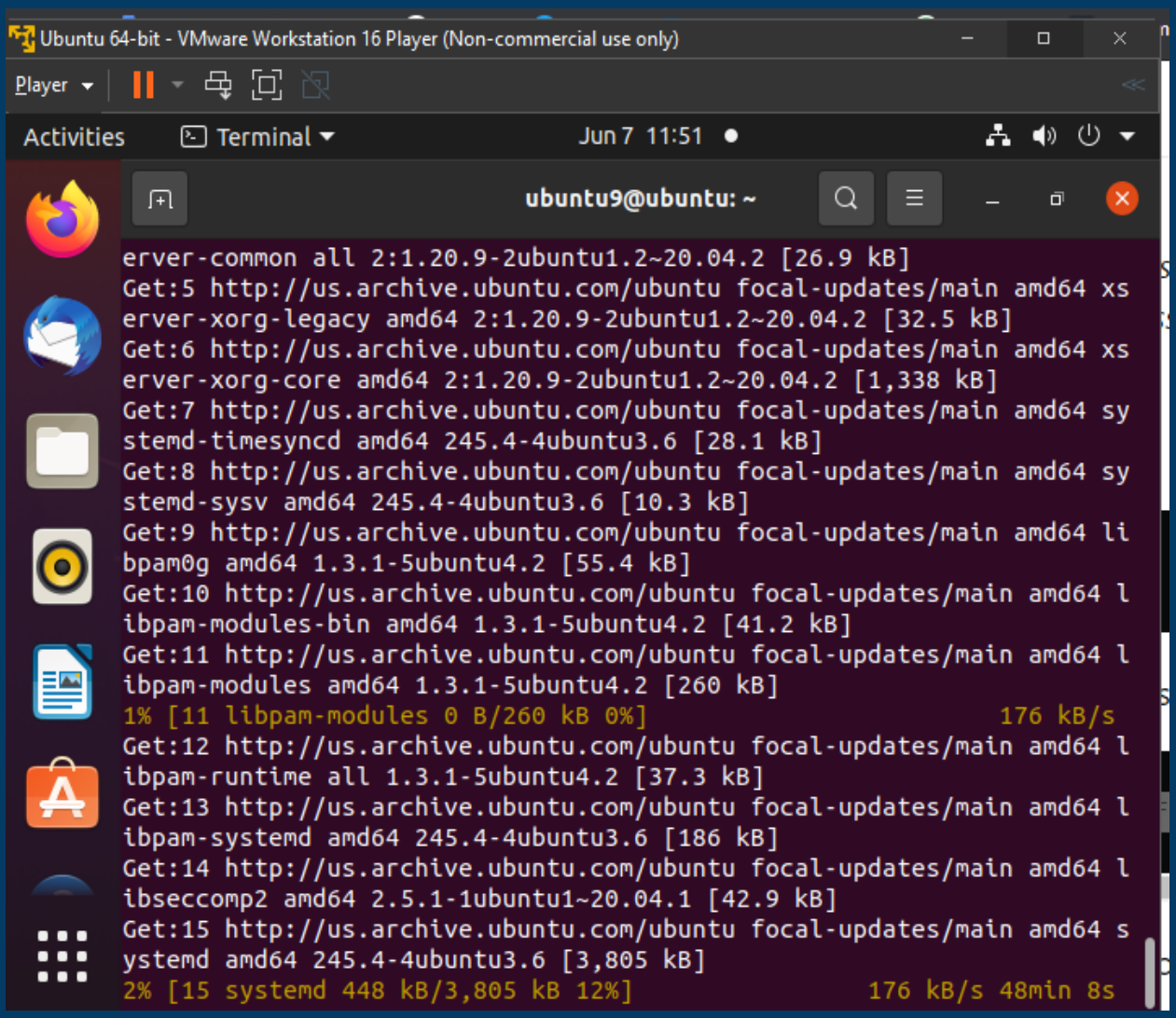
دينا احمد محمد

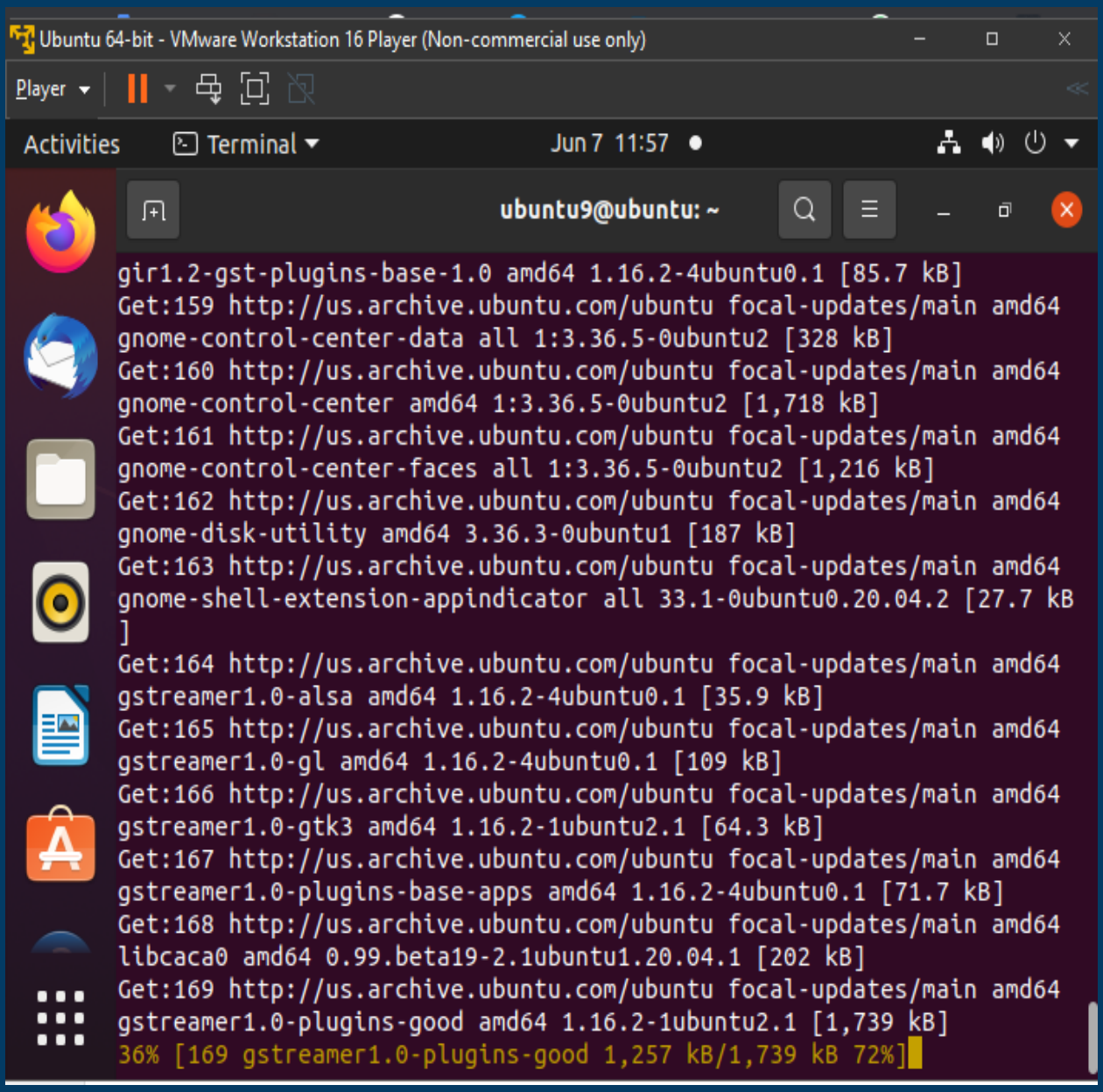
شيماء علي عبدالستار

group name:Dash

-*-The steps to add your system call the Linux OS kernel:

→update your operating system and Download and install the essential packages to compile kernels.

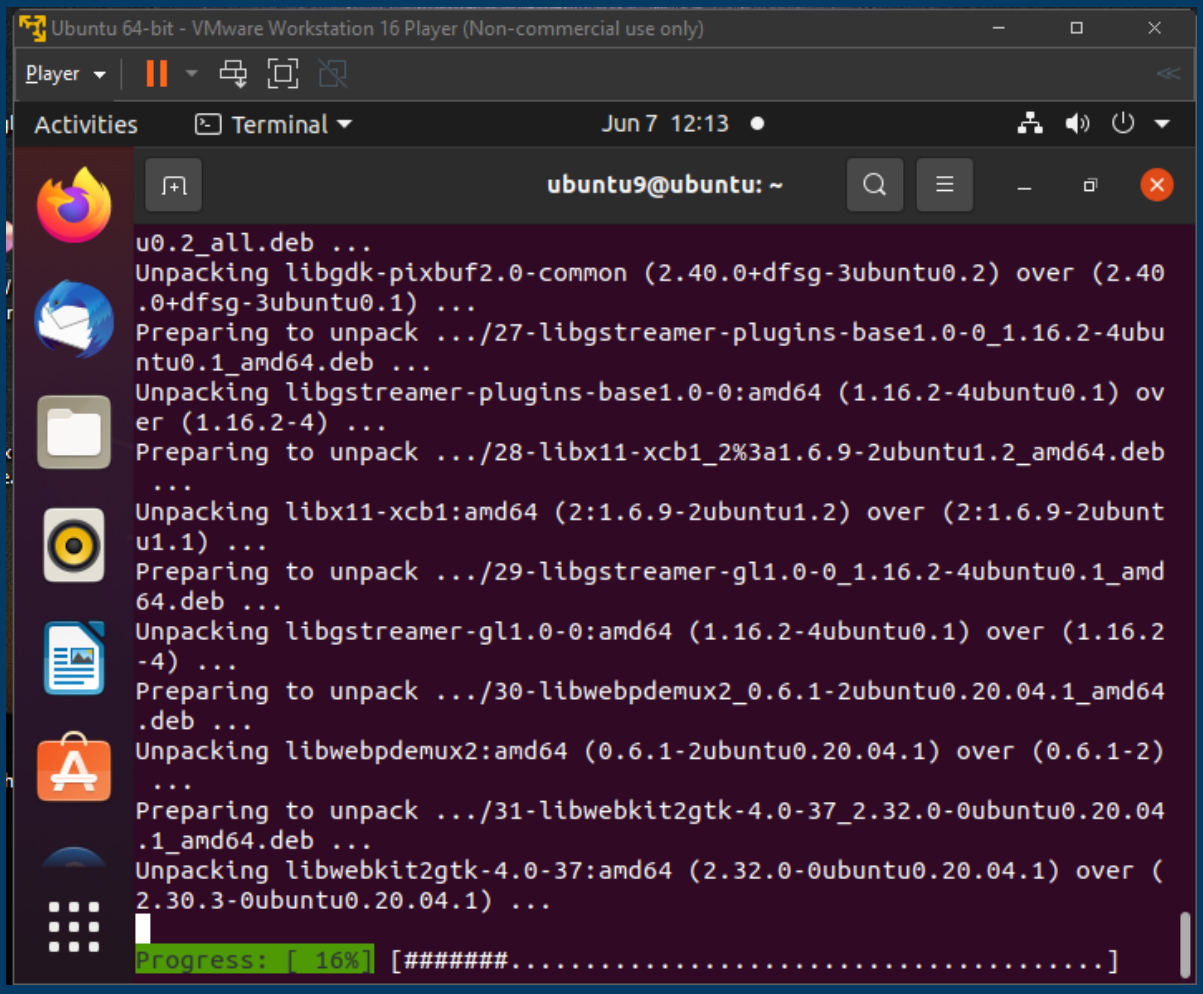


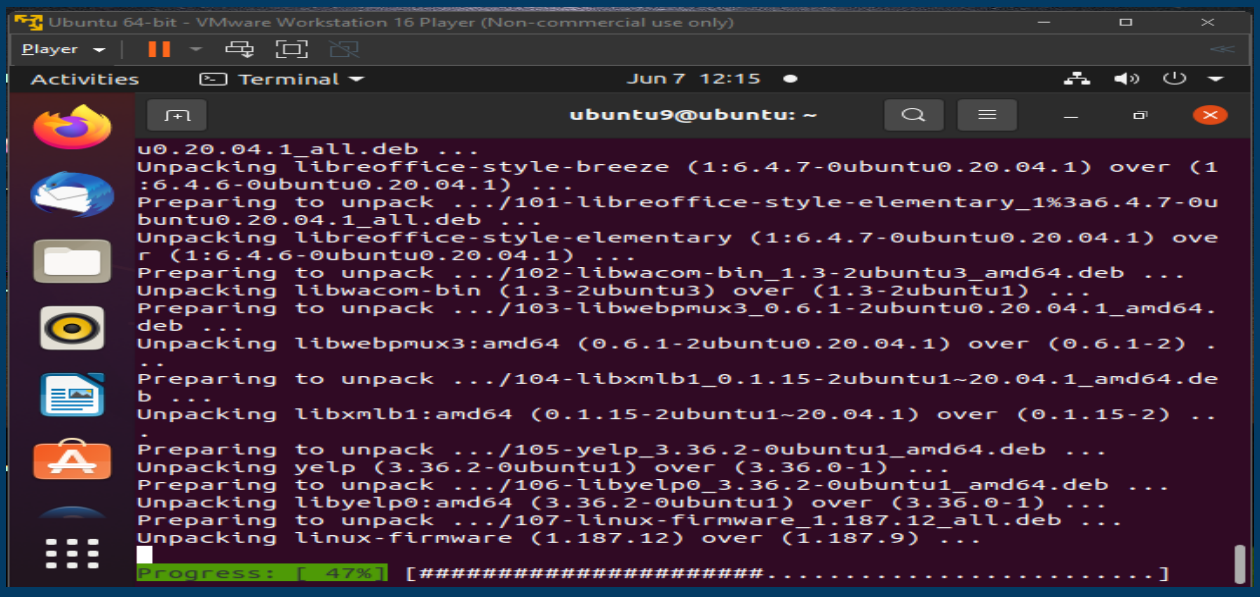


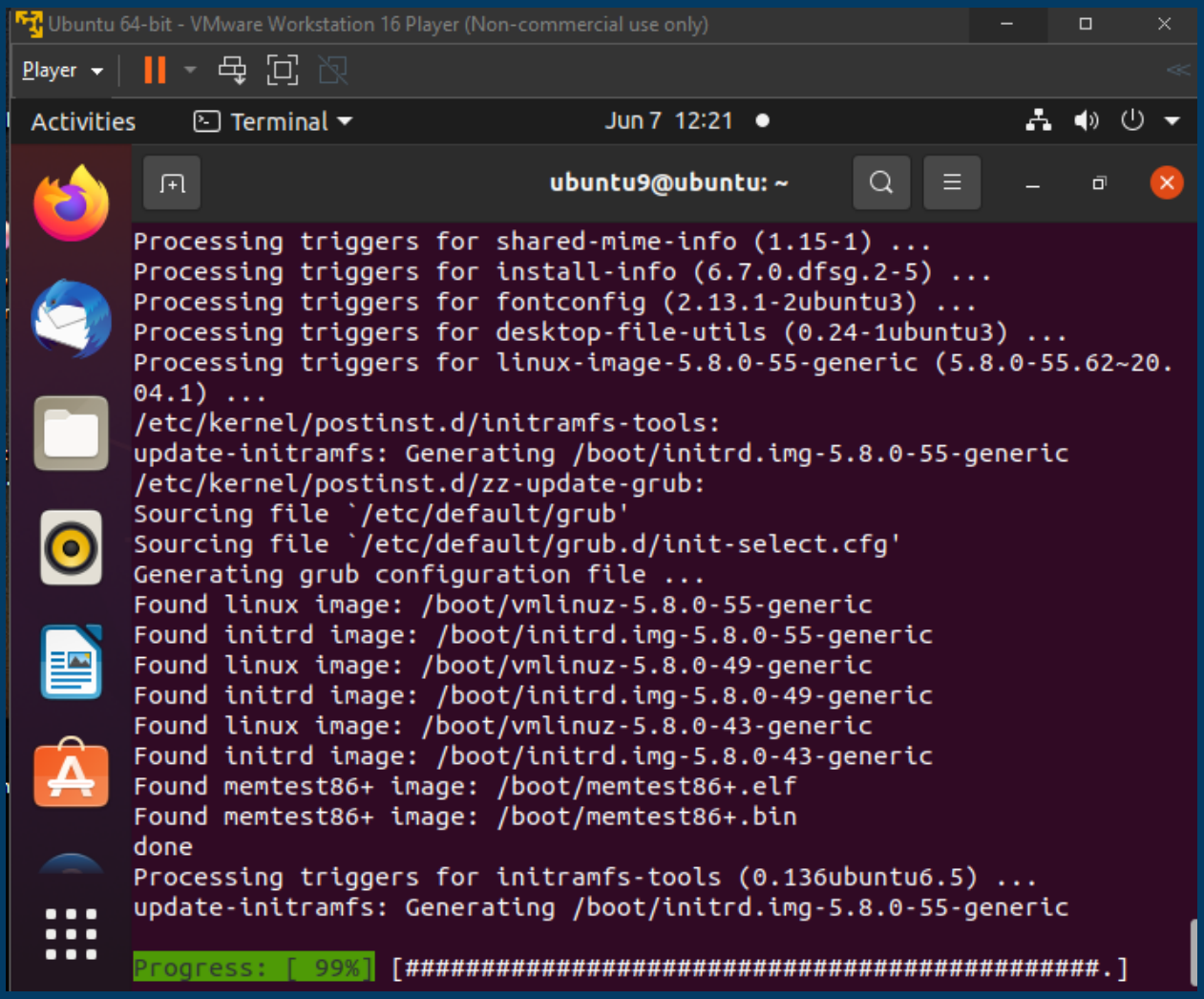
→Clean up your installed packages

Download the source code of the latest stable version of the Linux

kernel (which is 5.8.1) .

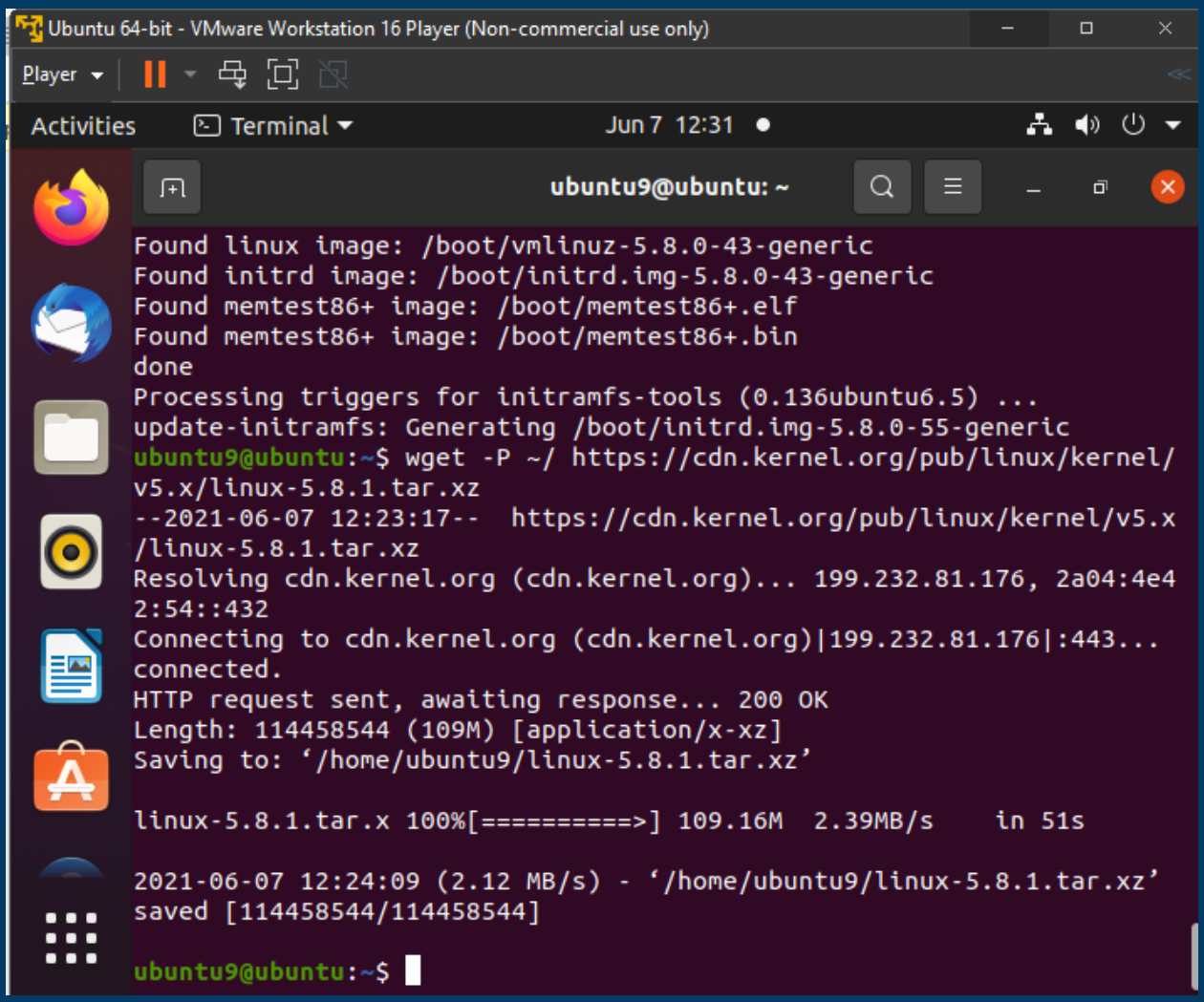


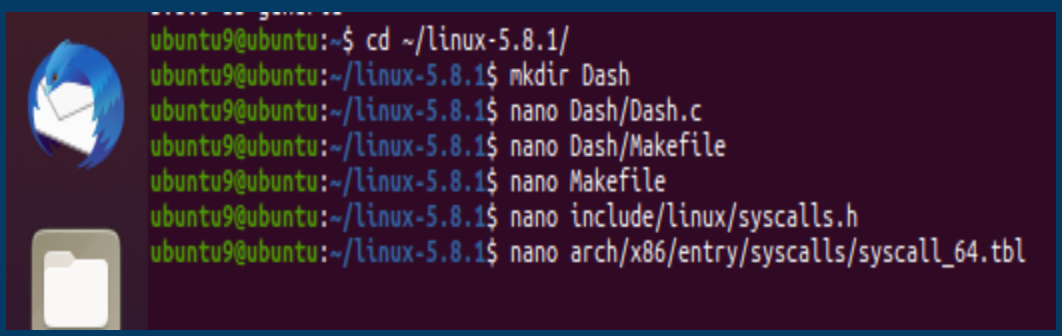




→you will write a basic system call in C and integrate it into the new kernel and Change your working directory to the root directory of the recently unpacked source code and Create the home directory of your system call. Decide a name for your

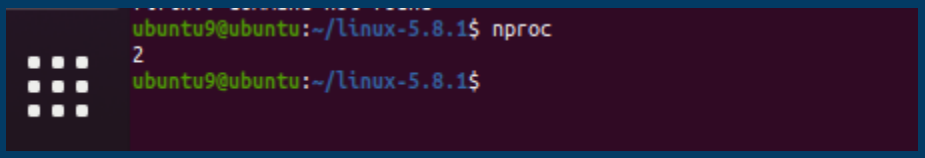
system call, and keep it consistent from this point onwards. I have chosen Dash





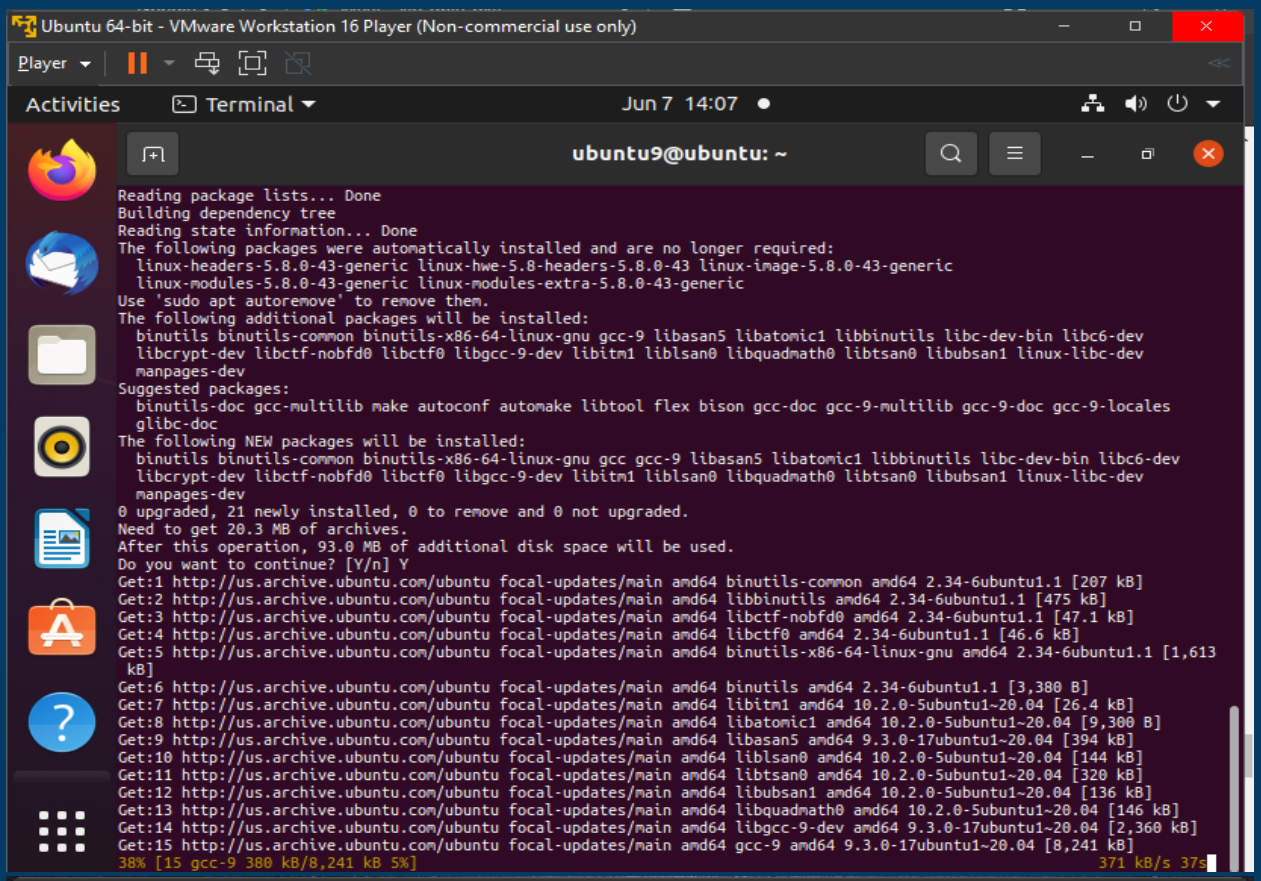
→you will install the new kernel and prepare your operating system to boot into it.

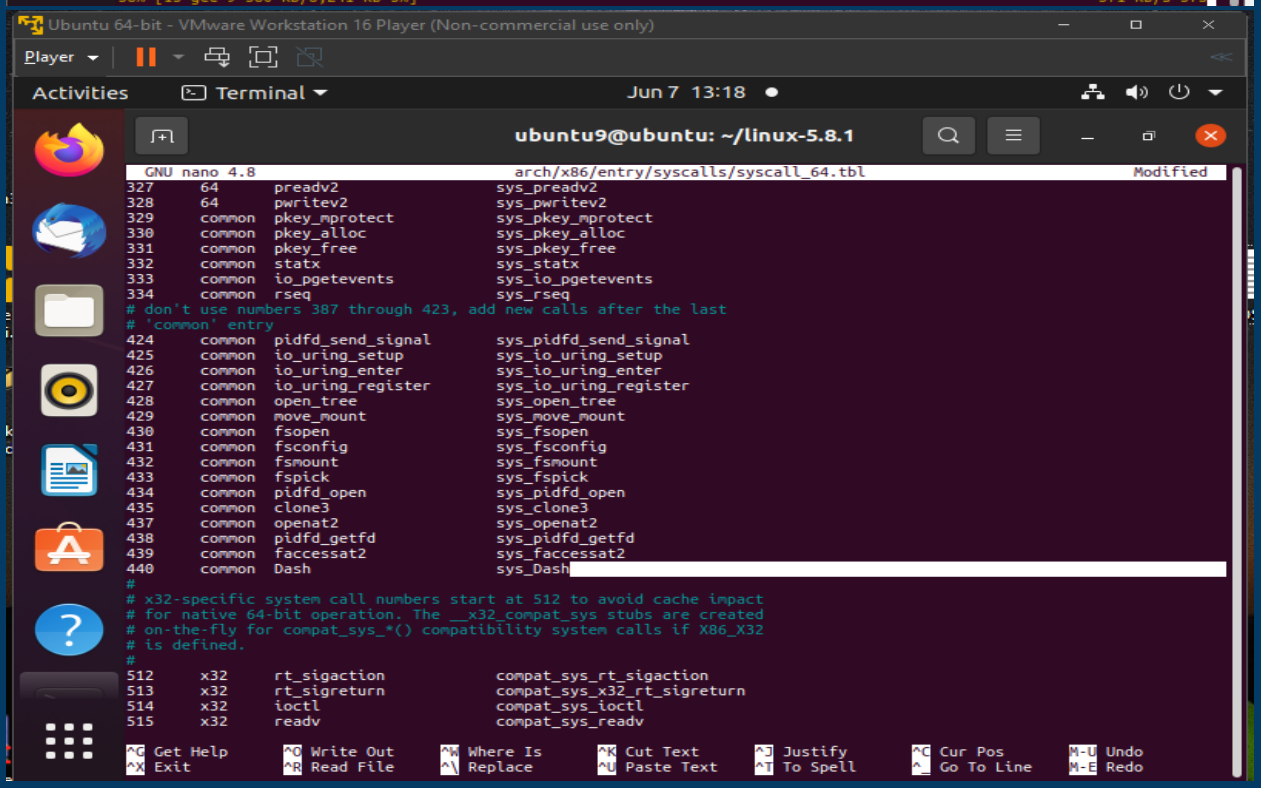
Find out how many logical cores you have.



Compile the kernel's source code and Prepare the installer of the kernel and Install the kernel.

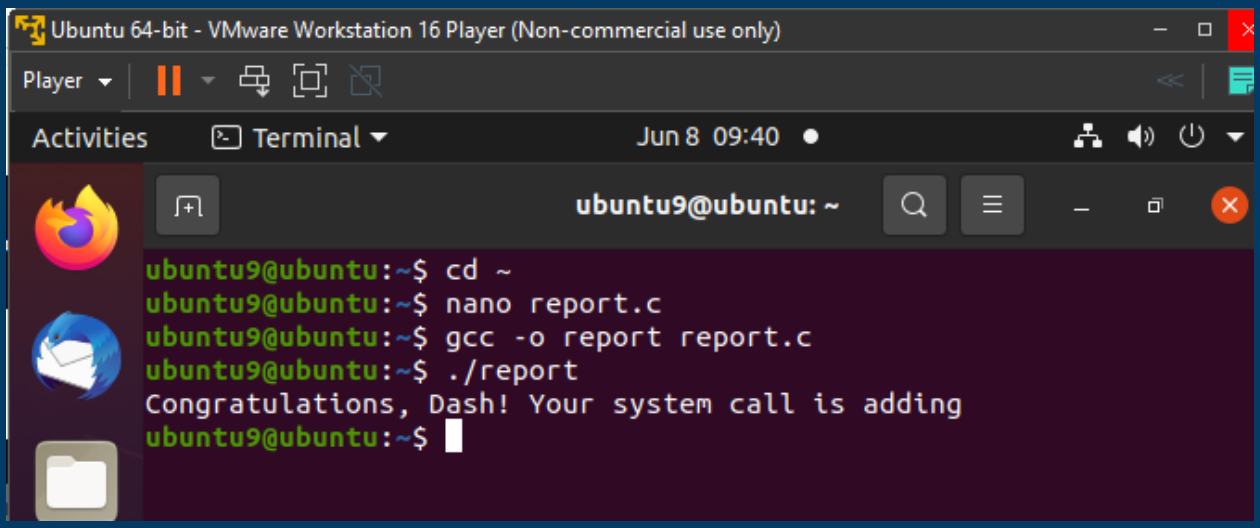
Update the bootloader of the operating system with the new kernel.


Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Terminal
Jun 7 14:07
ubuntu9@ubuntu: ~
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
linux-headers-5.8.0-43-generic linux-hwe-5.8-headers-5.8.0-43 linux-image-5.8.0-43-generic
linux-modules-5.8.0-43-generic linux-modules-extra-5.8.0-43-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
binutils binutils-common binutils-x86-64-linux-gnu gcc-9 libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev
libcrypt-dev libctf-nobfd0 libctf0 libgcc-9-dev libitm1 liblsan0 libquadmath0 libtsan0 libubsan1 linux-libc-dev
manpages-dev
Suggested packages:
binutils-doc gcc-multilib make autoconf automake libtool flex bison gcc-doc gcc-9-multilib gcc-9-doc gcc-9-locales
glibc-doc
The following NEW packages will be installed:
binutils binutils-common binutils-x86-64-linux-gnu gcc gcc-9 libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev
libcrypt-dev libctf-nobfd0 libctf0 libgcc-9-dev libitm1 liblsan0 libquadmath0 libtsan0 libubsan1 linux-libc-dev
manpages-dev
0 upgraded, 21 newly installed, 0 to remove and 0 not upgraded.
Need to get 20.3 MB of archives.
After this operation, 93.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 binutils-common amd64 2.34-6ubuntu1.1 [207 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libbinutils amd64 2.34-6ubuntu1.1 [475 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libctf-nobfd0 amd64 2.34-6ubuntu1.1 [47.1 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libctf0 amd64 2.34-6ubuntu1.1 [46.6 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 binutils-x86-64-linux-gnu amd64 2.34-6ubuntu1.1 [1,613 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 binutils amd64 2.34-6ubuntu1.1 [3,380 B]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libitm1 amd64 10.2.0-5ubuntu1~20.04 [26.4 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libatomic1 amd64 10.2.0-5ubuntu1~20.04 [9,300 B]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libasan5 amd64 9.3.0-17ubuntu1~20.04 [394 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 liblsan0 amd64 10.2.0-5ubuntu1~20.04 [144 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libtsan0 amd64 10.2.0-5ubuntu1~20.04 [320 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libubsan1 amd64 10.2.0-5ubuntu1~20.04 [136 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libquadmath0 amd64 10.2.0-5ubuntu1~20.04 [146 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libgcc-9-dev amd64 9.3.0-17ubuntu1~20.04 [2,360 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 gcc-9 amd64 9.3.0-17ubuntu1~20.04 [8,241 kB]
38% [15 gcc-9 380 kB/8,241 kB 5%]
371 kB/s 37s


Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Terminal
Jun 7 13:18
ubuntu9@ubuntu: ~/linux-5.8.1
GNU nano 4.8
arch/x86/entry/syscalls/syscall_64.tbl
Modified
327 64 preadv2 sys_preadv2
328 64 pwritev2 sys_pwritev2
329 common pkey_mprotect sys_pkey_mprotect
330 common pkey_alloc sys_pkey_alloc
331 common pkey_free sys_pkey_free
332 common statx sys_statx
333 common io_pgetevents sys_io_pgetevents
334 common rseq sys_rseq
don't use numbers 387 through 423, add new calls after the last
'common' entry
424 common pidfd_send_signal sys_pidfd_send_signal
425 common io_uring_setup sys_io_uring_setup
426 common io_uring_enter sys_io_uring_enter
427 common io_uring_register sys_io_uring_register
428 common open_tree sys_open_tree
429 common move_mount sys_move_mount
430 common fsopen sys_fsopen
431 common fsconfig sys_fsconfig
432 common fsmount sys_fsmount
433 common fspick sys_fspick
434 common pidfd_open sys_pidfd_open
435 common clone3 sys_clone3
437 common openat2 sys_openat2
438 common pidfd_getfd sys_pidfd_getfd
439 common faccessat2 sys_faccessat2
440 common Dash sys_Dash
#
x32-specific system call numbers start at 512 to avoid cache impact
for native 64-bit operation. The __x32_compat_sys stubs are created
on-the-fly for compat_sys_*() compatibility system calls if X86_X32
is defined.
#
512 x32 rt_sigaction compat_sys_rt_sigaction
513 x32 rt_sigreturn compat_sys_x32_rt_sigreturn
514 x32 ioctl compat_sys_ioctl
515 x32 readv compat_sys_readv
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo

→ you will write a C program to check whether your system call works or not. After that, you will see your system call in action. Check the version of your current kernel. It display (5.8.1)

Change your working directory to your home directory and Create a C file to generate a report of the success or failure of your system call.



*/*My system include CPU cores =2, RAM capacity 2GB and Kernel version 5.8.1