NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

PROGRAM: SOFTWARE ENGINEERING

OPERATING SYSTEMS LAB

ASSIGNMENT-03

SUBMITTED BY:

Name: Ahmed Ali

Roll No: 22P-9318

Section: BS(SE)-5B

INSTRUCTOR NAME: Sir Saad Ahmad

A DEPARTMENT OF COMPUTER SCIENCE

Purpose of Each Command:

SHA256SUM:

SHA256SUM is a hash value generated using the SHA-256 algorithm it is used to verify the integrity of a file, ensuring it has not been corrupted during download

By comparing the computed hash of a downloaded file with a provided hash we can confirm the file is authenticity and integrity

Shebang & Script Setup

- #!/bin/bash: Specifies the script should run using the bash shell
- set -e: Exits the script if any command returns error message
- set -o pipefail: Ensures the script exits if any command in a pipeline fails

Variables

- KERNEL_URL, KERNEL_ARCHIVE, KERNEL_DIR, SHA256SUM_URL: Store the URLs and file names for the Linux kernel and associated checksum file

Log Function

- log(): A custom function to output messages It helps maintain consistent formatting of messages.

- I have used log more instead of echo because by using log we can maintain a backup file of what we are doing.

Installing Dependencies:

- build-essential: Includes tools like gcc and make
- libncurses-dev: Enables text-based UI for kernel configuration
- bison & flex: For parsing kernel source code
- libssl-dev: Adds support for cryptographic features
- wget: For downloading files
- bc: Used in the kernel build process
- xz-utils: For extracting .xz files
- gnupg: Verifies signed files
- libelf-dev: Needed for ELF file handling in the kernel

Downloading Kernel

```
wget -O $KERNEL_ARCHIVE $KERNEL_URL
wget -O sha256sums.asc $SHA256SUM_URL
```

Downloads the Linux kernel source archive and its associated checksum file from google

Verifying Integrity ensures that a file or data has not been modified

```
grep $(basename $KERNEL_ARCHIVE) sha256sums.asc | sha256sum -c -
```

- **grep $(basename $KERNEL_ARCHIVE) sha256sums.asc:** extracts the hash for the kernel file from the checksum file.
- **sha256sum -c -**: compares the extracted hash with the calculated hash of the downloaded kernel.

RESULT: ensures the file was downloaded correctly and matches the expected hash table

Extracting Kernel

```
tar -xf $KERNEL_ARCHIVE
```

Extracts the downloaded kernel source file

Changing Directory

```
cd $KERNEL_DIR
```

Moving into the kernel directory

Kernel Configuration

```
make defconfig
```

Generates a default configuration for the kernel

Compiling the Kernel

```
make -j$(nproc)
```

Compiles the kernel using all available CPU cores (nproc gives the number of cores)

Compiling Kernel Modules

```
make modules
```

Compiles kernel modules

Installing Kernel and Modules

sudo make modules_install: Installs the compiled kernel modules

sudo make install: Installs the compiled kernel and updates boot files

Updating GRUB

```
sudo update-grub
```

Updates the GRUB bootloader to include the new kernel

Cleanup (removal of temporary files)

```
rm -rf $KERNEL_ARCHIVE $KERNEL_DIR sha256sums.asc
```

Deletes temporary files and directoris created during the kernel installation process

Reboot

```
read -p "do U want to reboot to apply the new kernel? (y/n): "
```

```
REBOOT
```

Ask the user to reboot the system to apply the new kernel

ADVANTAGES OF LOG:

If you need to redirect or modify outputs (e.g., log to a file), you only update the log function.

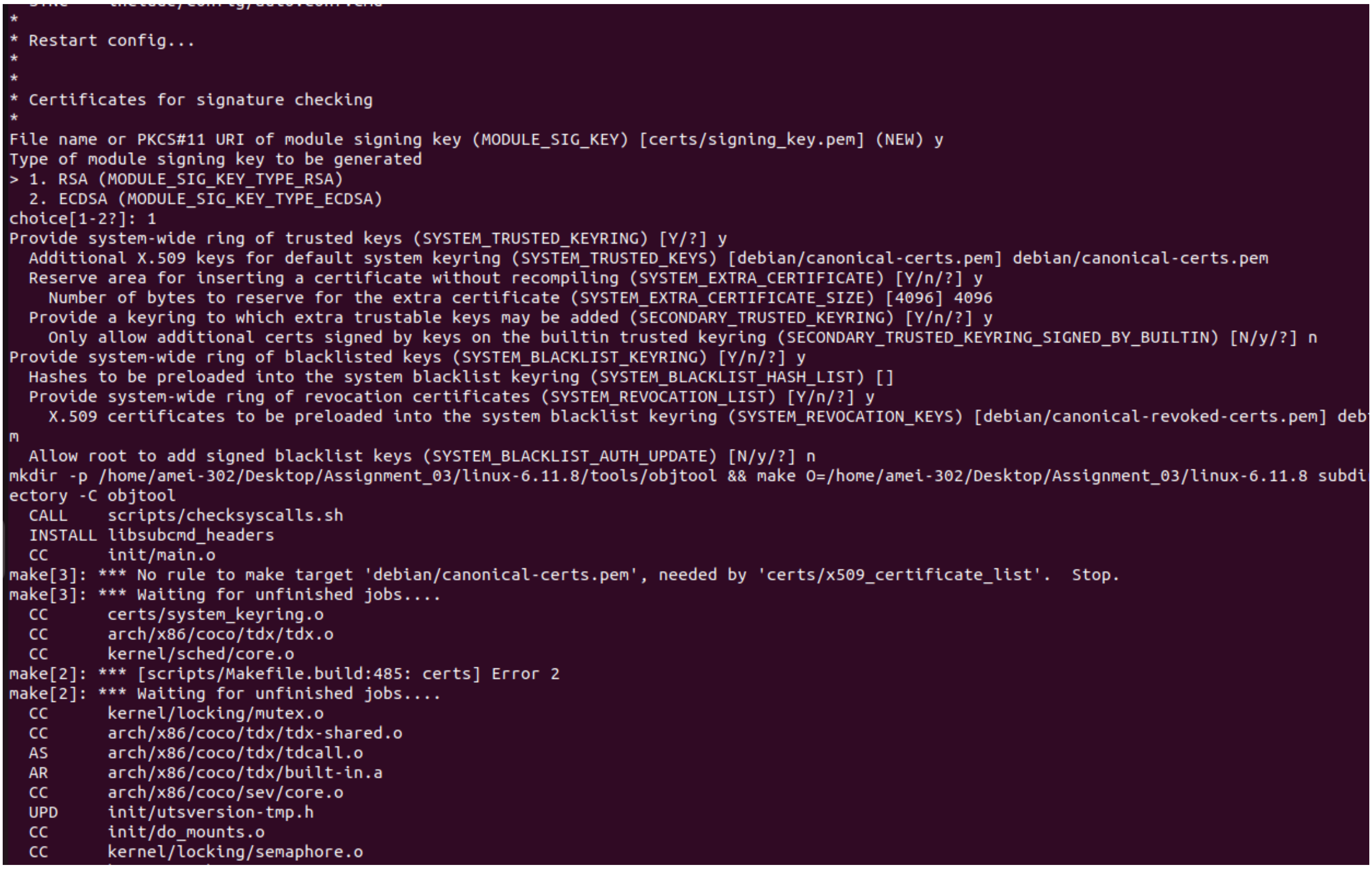
CHALLENGES:

The Challenges I faced are during customized configuration of kernel while it is in compilation process, it was asking for certificates for signature checking.

Even I have disabled **CONFIG_MODULE_SIG** but it was asking for the key

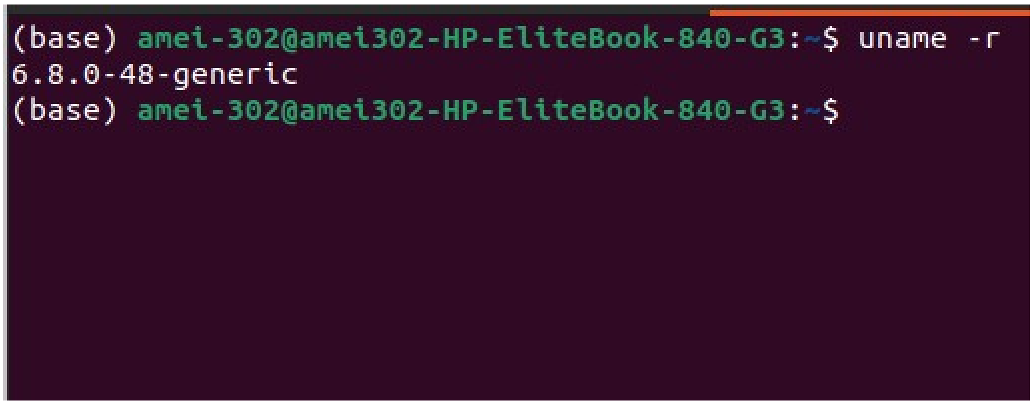
Then I remember that there is no restriction for default configuration so, therefore I used **make defconfig command** which compiles the kernel using default settings of configuration

Image is attached below you can see make[2] and make[3] after which compilation started but ends with error.



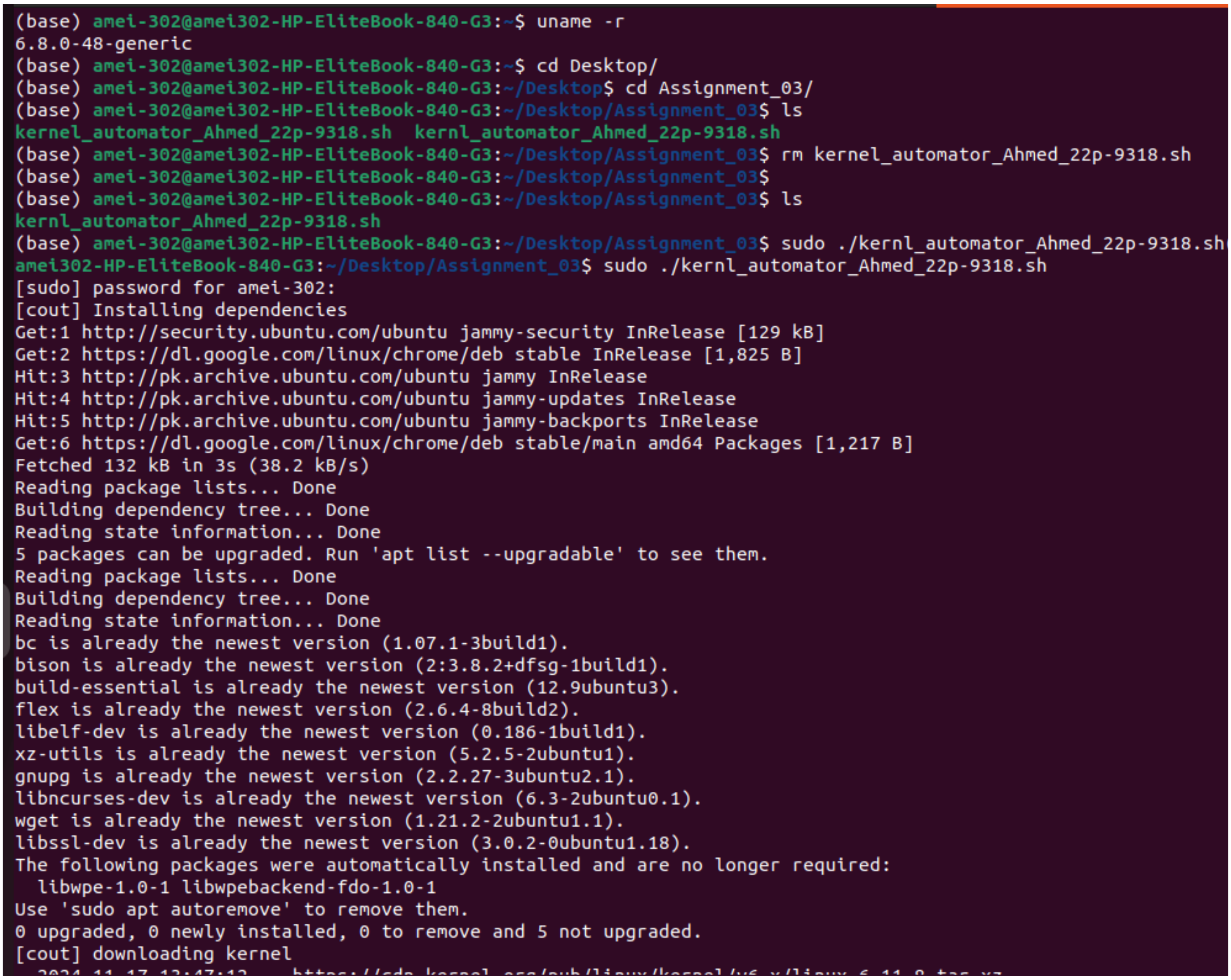
```
  SYNC    include/config/auto.conf.cmd
*
* Restart config...
*
*
* Certificates for signature checking
*
File name or PKCS#11 URI of module signing key (MODULE_SIG_KEY) [certs/signing_key.pem] (NEW) y
Type of module signing key to be generated
> 1. RSA (MODULE_SIG_KEY_TYPE_RSA)
  2. ECDSA (MODULE_SIG_KEY_TYPE_ECDSA)
choice[1-2?]: 1
Provide system-wide ring of trusted keys (SYSTEM_TRUSTED_KEYRING) [Y/?] y
  Additional X.509 keys for default system keyring (SYSTEM_TRUSTED_KEYS) [debian/canonical-certs.pem] debian/canonical-certs.pem
  Reserve area for inserting a certificate without recompiling (SYSTEM_EXTRA_CERTIFICATE) [Y/n/?] y
    Number of bytes to reserve for the extra certificate (SYSTEM_EXTRA_CERTIFICATE_SIZE) [4096] 4096
  Provide a keyring to which extra trustable keys may be added (SECONDARY_TRUSTED_KEYRING) [Y/n/?] y
    Only allow additional certs signed by keys on the builtin trusted keyring (SECONDARY_TRUSTED_KEYRING_SIGNED_BY_BUILTIN) [N/y/?] n
Provide system-wide ring of blacklisted keys (SYSTEM_BLACKLIST_KEYRING) [Y/n/?] y
  Hashes to be preloaded into the system blacklist keyring (SYSTEM_BLACKLIST_HASH_LIST) []
  Provide system-wide ring of revocation certificates (SYSTEM_REVOCATION_LIST) [Y/n/?] y
    X.509 certificates to be preloaded into the system blacklist keyring (SYSTEM_REVOCATION_KEYS) [debian/canonical-revoked-certs.pem] deb
m
  Allow root to add signed blacklist keys (SYSTEM_BLACKLIST_AUTH_UPDATE) [N/y/?] n
mkdir -p /home/amei-302/Desktop/Assignment_03/linux-6.11.8/tools/objtool && make O=/home/amei-302/Desktop/Assignment_03/linux-6.11.8 subdi
ectory -C objtool
  CALL    scripts/checksyscalls.sh
  INSTALL libsubcmd_headers
  CC      init/main.o
make[3]: *** No rule to make target 'debian/canonical-certs.pem', needed by 'certs/x509_certificate_list'.  Stop.
make[3]: *** Waiting for unfinished jobs....
  CC      certs/system_keyring.o
  CC      arch/x86/coco/tdx/tdx.o
  CC      kernel/sched/core.o
make[2]: *** [scripts/Makefile.build:485: certs] Error 2
make[2]: *** Waiting for unfinished jobs....
  CC      kernel/locking/mutex.o
  CC      arch/x86/coco/tdx/tdx-shared.o
  AS      arch/x86/coco/tdx/tdcall.o
  AR      arch/x86/coco/tdx/built-in.a
  CC      arch/x86/coco/sev/core.o
  UPD     init/utsversion-tmp.h
  CC      init/do_mounts.o
  CC      kernel/locking/semaphore.o
```

Current kernel:



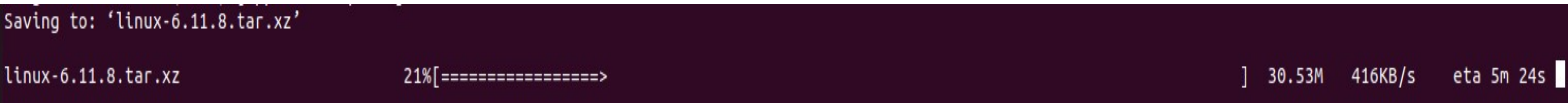
```
(base) amei-302@amei302-HP-EliteBook-840-G3:~$ uname -r
6.8.0-48-generic
(base) amei-302@amei302-HP-EliteBook-840-G3:~$
```

Now RUNNING SCRIPT:



```
(base) amei-302@amei302-HP-EliteBook-840-G3:~$ uname -r
6.8.0-48-generic
(base) amei-302@amei302-HP-EliteBook-840-G3:~$ cd Desktop/
(base) amei-302@amei302-HP-EliteBook-840-G3:~/Desktop$ cd Assignment_03/
(base) amei-302@amei302-HP-EliteBook-840-G3:~/Desktop/Assignment_03$ ls
kernel_automator_Ahmed_22p-9318.sh  kernl_automator_Ahmed_22p-9318.sh
(base) amei-302@amei302-HP-EliteBook-840-G3:~/Desktop/Assignment_03$ rm kernel_automator_Ahmed_22p-9318.sh
(base) amei-302@amei302-HP-EliteBook-840-G3:~/Desktop/Assignment_03$
(base) amei-302@amei302-HP-EliteBook-840-G3:~/Desktop/Assignment_03$ ls
kernl_automator_Ahmed_22p-9318.sh
(base) amei-302@amei302-HP-EliteBook-840-G3:~/Desktop/Assignment_03$ sudo ./kernl_automator_Ahmed_22p-9318.sh
amei302-HP-EliteBook-840-G3:~/Desktop/Assignment_03$ sudo ./kernl_automator_Ahmed_22p-9318.sh
[sudo] password for amei-302:
[cout] Installing dependencies
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Hit:3 http://pk.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://pk.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://pk.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:6 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,217 B]
Fetched 132 kB in 3s (38.2 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
5 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bc is already the newest version (1.07.1-3build1).
bison is already the newest version (2:3.8.2+dfsg-1build1).
build-essential is already the newest version (12.9ubuntu3).
flex is already the newest version (2.6.4-8build2).
libelf-dev is already the newest version (0.186-1build1).
xz-utils is already the newest version (5.2.5-2ubuntu1).
gnupg is already the newest version (2.2.27-3ubuntu2.1).
libncurses-dev is already the newest version (6.3-2ubuntu0.1).
wget is already the newest version (1.21.2-2ubuntu1.1).
libssl-dev is already the newest version (3.0.2-0ubuntu1.18).
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
[cout] downloading kernel
--2024-11-17 12:47:12--  https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.11.8.tar.xz
```

Downloading kernel:



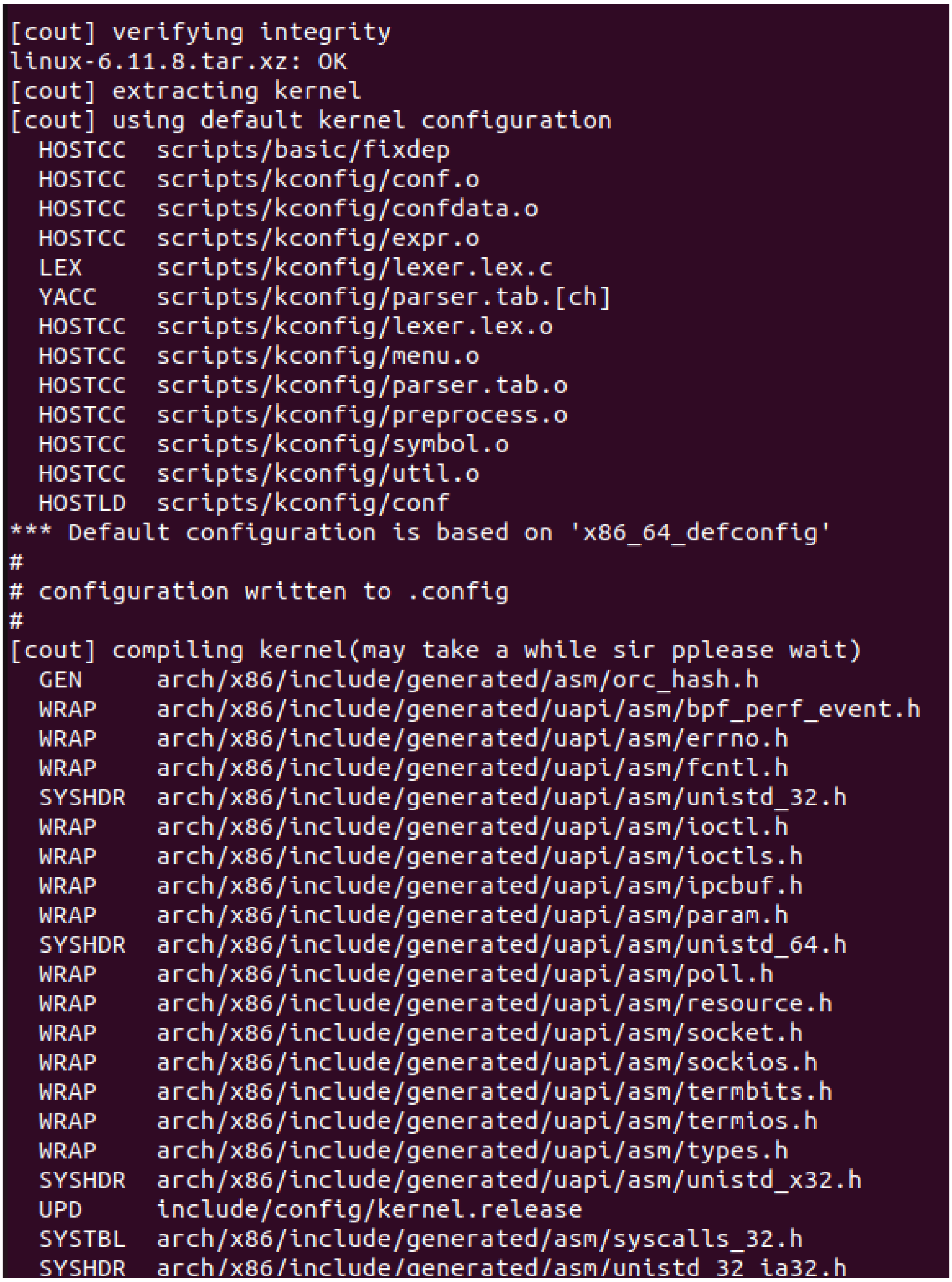
```
Saving to: ‘linux-6.11.8.tar.xz’

linux-6.11.8.tar.xz                    21%[=================>                                                              ]  30.53M   416KB/s    eta 5m 24s
```

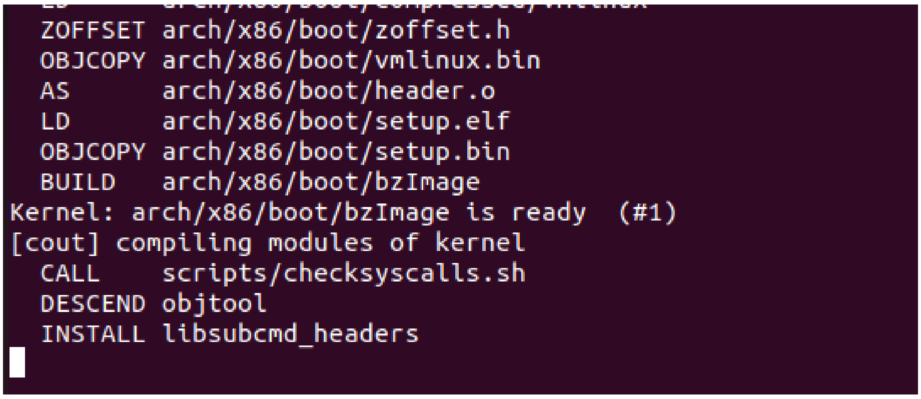
Integrity Verification: ensures that a file or data has not been modified

Configuring and Compilation:

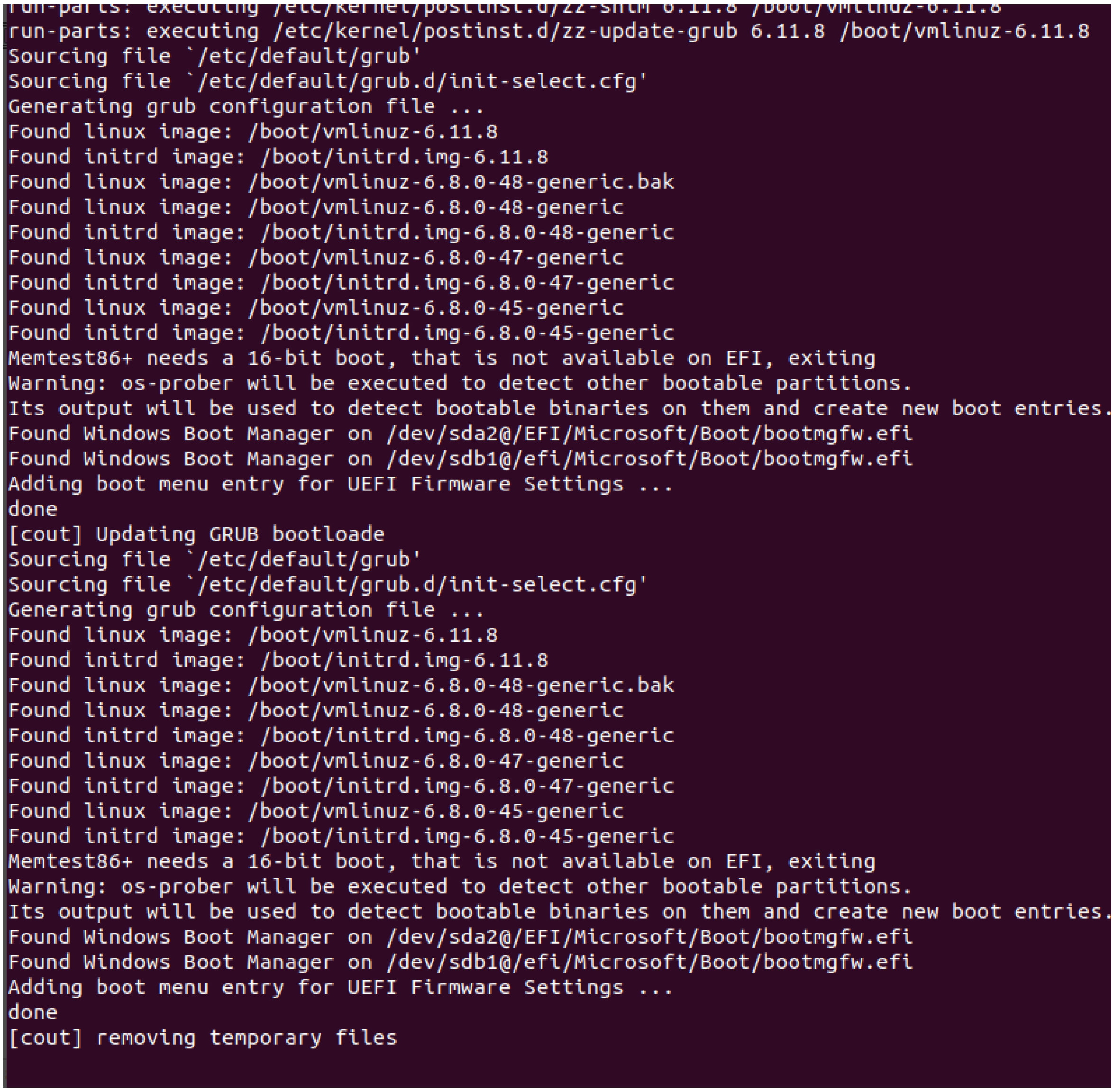
IMAGE IS ATTACHED BELOW:



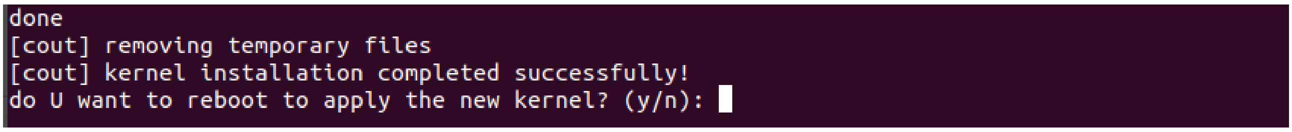
```
[cout] verifying integrity
linux-6.11.8.tar.xz: OK
[cout] extracting kernel
[cout] using default kernel configuration
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  HOSTCC  scripts/kconfig/confdata.o
  HOSTCC  scripts/kconfig/expr.o
  LEX     scripts/kconfig/lexer.lex.c
  YACC    scripts/kconfig/parser.tab.[ch]
  HOSTCC  scripts/kconfig/lexer.lex.o
  HOSTCC  scripts/kconfig/menu.o
  HOSTCC  scripts/kconfig/parser.tab.o
  HOSTCC  scripts/kconfig/preprocess.o
  HOSTCC  scripts/kconfig/symbol.o
  HOSTCC  scripts/kconfig/util.o
  HOSTLD  scripts/kconfig/conf
*** Default configuration is based on 'x86_64_defconfig'
#
# configuration written to .config
#
[cout] compiling kernel(may take a while sir pplease wait)
  GEN     arch/x86/include/generated/asm/orc_hash.h
  WRAP    arch/x86/include/generated/uapi/asm/bpf_perf_event.h
  WRAP    arch/x86/include/generated/uapi/asm/errno.h
  WRAP    arch/x86/include/generated/uapi/asm/fcntl.h
  SYSHDR  arch/x86/include/generated/uapi/asm/unistd_32.h
  WRAP    arch/x86/include/generated/uapi/asm/ioctl.h
  WRAP    arch/x86/include/generated/uapi/asm/ioctls.h
  WRAP    arch/x86/include/generated/uapi/asm/ipcbuf.h
  WRAP    arch/x86/include/generated/uapi/asm/param.h
  SYSHDR  arch/x86/include/generated/uapi/asm/unistd_64.h
  WRAP    arch/x86/include/generated/uapi/asm/poll.h
  WRAP    arch/x86/include/generated/uapi/asm/resource.h
  WRAP    arch/x86/include/generated/uapi/asm/socket.h
  WRAP    arch/x86/include/generated/uapi/asm/sockios.h
  WRAP    arch/x86/include/generated/uapi/asm/termbits.h
  WRAP    arch/x86/include/generated/uapi/asm/termios.h
  WRAP    arch/x86/include/generated/uapi/asm/types.h
  SYSHDR  arch/x86/include/generated/uapi/asm/unistd_x32.h
  UPD     include/config/kernel.release
  SYSTBL  arch/x86/include/generated/asm/syscalls_32.h
  SYSHDR  arch/x86/include/generated/asm/unistd_32_ia32.h
```



```
  ZOFFSET arch/x86/boot/zoffset.h
  OBJCOPY arch/x86/boot/vmlinux.bin
  AS      arch/x86/boot/header.o
  LD      arch/x86/boot/setup.elf
  OBJCOPY arch/x86/boot/setup.bin
  BUILD   arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready  (#1)
[cout] compiling modules of kernel
  CALL    scripts/checksyscalls.sh
  DESCEND objtool
  INSTALL libsubcmd_headers
```

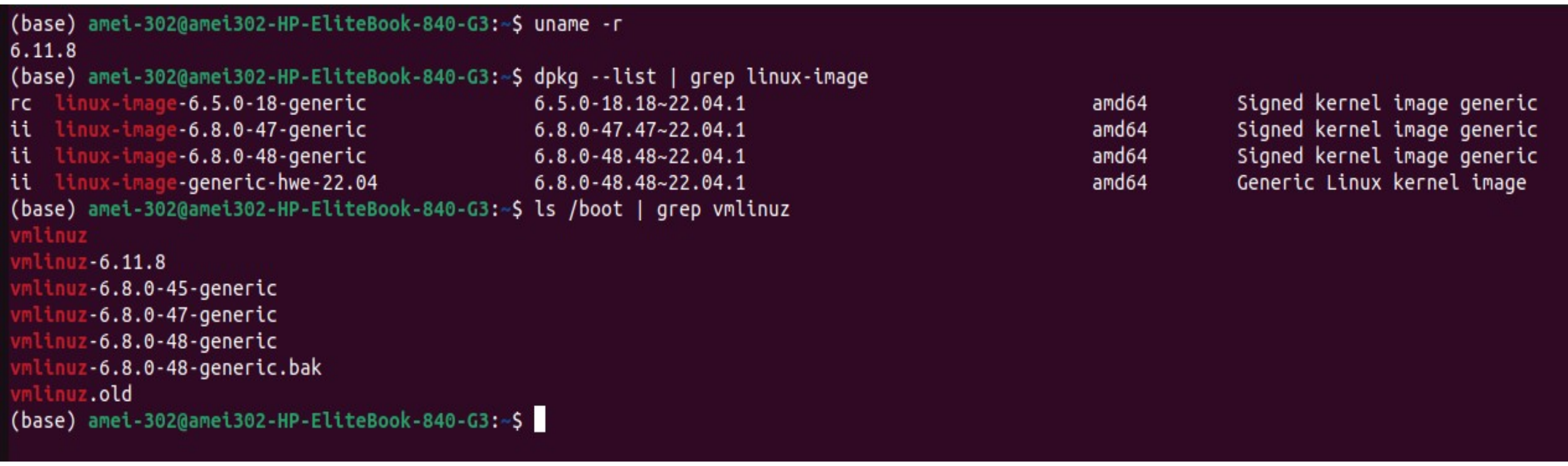


```
run-parts: executing /etc/kernel/postinst.d/zz-shim 6.11.8 /boot/vmlinuz-6.11.8
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.11.8 /boot/vmlinuz-6.11.8
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.11.8
Found initrd image: /boot/initrd.img-6.11.8
Found linux image: /boot/vmlinuz-6.8.0-48-generic.bak
Found linux image: /boot/vmlinuz-6.8.0-48-generic
Found initrd image: /boot/initrd.img-6.8.0-48-generic
Found linux image: /boot/vmlinuz-6.8.0-47-generic
Found initrd image: /boot/initrd.img-6.8.0-47-generic
Found linux image: /boot/vmlinuz-6.8.0-45-generic
Found initrd image: /boot/initrd.img-6.8.0-45-generic
Memtest86+ needs a 16-bit boot, that is not available on EFI, exiting
Warning: os-prober will be executed to detect other bootable partitions.
Its output will be used to detect bootable binaries on them and create new boot entries.
Found Windows Boot Manager on /dev/sda2@/EFI/Microsoft/Boot/bootmgfw.efi
Found Windows Boot Manager on /dev/sdb1@/efi/Microsoft/Boot/bootmgfw.efi
Adding boot menu entry for UEFI Firmware Settings ...
done
[cout] Updating GRUB bootloade
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.11.8
Found initrd image: /boot/initrd.img-6.11.8
Found linux image: /boot/vmlinuz-6.8.0-48-generic.bak
Found linux image: /boot/vmlinuz-6.8.0-48-generic
Found initrd image: /boot/initrd.img-6.8.0-48-generic
Found linux image: /boot/vmlinuz-6.8.0-47-generic
Found initrd image: /boot/initrd.img-6.8.0-47-generic
Found linux image: /boot/vmlinuz-6.8.0-45-generic
Found initrd image: /boot/initrd.img-6.8.0-45-generic
Memtest86+ needs a 16-bit boot, that is not available on EFI, exiting
Warning: os-prober will be executed to detect other bootable partitions.
Its output will be used to detect bootable binaries on them and create new boot entries.
Found Windows Boot Manager on /dev/sda2@/EFI/Microsoft/Boot/bootmgfw.efi
Found Windows Boot Manager on /dev/sdb1@/efi/Microsoft/Boot/bootmgfw.efi
Adding boot menu entry for UEFI Firmware Settings ...
done
[cout] removing temporary files
```



```
done
[cout] removing temporary files
[cout] kernel installation completed successfully!
do U want to reboot to apply the new kernel? (y/n):
```

Enter y and press enter key here, The system will reboot and the newly installed kernel will run



```
(base) amei-302@amei302-HP-EliteBook-840-G3:~$ uname -r
6.11.8
(base) amei-302@amei302-HP-EliteBook-840-G3:~$ dpkg --list | grep linux-image
rc  linux-image-6.5.0-18-generic          6.5.0-18.18~22.04.1                         amd64        Signed kernel image generic
ii  linux-image-6.8.0-47-generic          6.8.0-47.47~22.04.1                         amd64        Signed kernel image generic
ii  linux-image-6.8.0-48-generic          6.8.0-48.48~22.04.1                         amd64        Signed kernel image generic
ii  linux-image-generic-hwe-22.04         6.8.0-48.48~22.04.1                         amd64        Generic Linux kernel image
(base) amei-302@amei302-HP-EliteBook-840-G3:~$ ls /boot | grep vmlinuz
vmlinuz
vmlinuz-6.11.8
vmlinuz-6.8.0-45-generic
vmlinuz-6.8.0-47-generic
vmlinuz-6.8.0-48-generic
vmlinuz-6.8.0-48-generic.bak
vmlinuz.old
(base) amei-302@amei302-HP-EliteBook-840-G3:~$
```