

*Advanced OS*

*Lab 6*



# The Linux Kernel

 kheder hassoun

 Eng.Reem\_Hasan 

- I already Downloaded The most Stable Linux Kernel (6.9.3) on windows OS

So, let's just paste it in my upuntu Linux 😁

**The problem is I don't have User interface in my upuntu so to copy the file from windows to VM I used sftp (Secure file transfer Protocol)**

```
C:\Users\kheder\Downloads>sftp kheder@192.168.184.132
kheder@192.168.184.132's password:
Connected to 192.168.184.132.
sftp> put linux-6.9.3.tar.xz
Uploading linux-6.9.3.tar.xz to /home/kheder/linux-6.9.3.tar.xz
linux-6.9.3.tar.xz
sftp>
```

- Now let's copy file from **home/kheder** To **usr/src** :

```
root@kheder:/home/kheder# ls
docker linux-6.9.3.tar.xz
root@kheder:/home/kheder# sudo cp linux-6.9.3.tar.xz /usr/src
```

- Look how much it's pretty 😊

```
root@kheder:/usr/src# ls -l
total 140680
-rw-r--r-- 1 root root 144036552 Jun 22 07:20 linux-6.9.3.tar.xz
drwxr-xr-x 25 root root 4096 Jun 4 00:13 linux-headers-5.15.0-107
drwxr-xr-x 7 root root 4096 Jun 4 00:14 linux-headers-5.15.0-107-generic
drwxr-xr-x 25 root root 4096 Jun 22 07:07 linux-headers-5.15.0-112
drwxr-xr-x 7 root root 4096 Jun 22 07:07 linux-headers-5.15.0-112-generic
root@kheder:/usr/src#
```

- To continue installing we have to install some packages (it doesn't exist in my upuntu 64 😞)

⚠️ But firstly Because I can't Past any command in my stupid Upuntu I use **ssh** to connect it from my windows 😞 😞

```
kheder@kheder:~$ sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc flex libe lf-dev b
ison
[sudo] password for kheder:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'libncurses-dev' instead of 'ncurses-dev'
E: Unable to locate package libe
E: Unable to locate package lf-dev
kheder@kheder:~$
```

- Now Let's see the old Kernel Version 🖥️

```
kheder@kheder:~$ hostnamectl
Static hostname: kheder
Icon name: computer-vm
Chassis: vm
Machine ID: 29d70d5783bc46108c4626204d7d529c
Boot ID: e081547a7b564d1294c2f2ea1c95992e
Virtualization: vmware
Operating System: Ubuntu 22.04.4 LTS
Kernel: Linux 5.15.0-107-generic
Architecture: x86-64
Hardware Vendor: VMware, Inc.
Hardware Model: VMware Virtual Platform
kheder@kheder:~$
```

- Extract the new kernel file

```
kheder@kheder:~$ tar xvf linux-6.9.3.tar.xz
```

```
kheder@kheder:~$ cd /usr/src/linux-6.9.3
kheder@kheder:~/usr/src/linux-6.9.3$ find . -type f | grep -E '\.(c|h)$' | sort
./usr/include/
./usr/include/.gitignore
./usr/include/Makefile
./usr/include/headers_check.pl
./usr/initramfs_data.5
./virt/
./virt/Makefile
./virt/kvm/
./virt/kvm/Kconfig
./virt/kvm/Makefile.kvm
./virt/kvm/async_pf.c
./virt/kvm/async_pf.h
./virt/kvm/binary_stats.c
./virt/kvm/coalesced_mmio.c
./virt/kvm/coalesced_mmio.h
./virt/kvm/dirty_ring.c
./virt/kvm/eventfd.c
./virt/kvm/guest_memfd.c
./virt/kvm/irqchip.c
./virt/kvm/kvm_main.c
./virt/kvm/kvm_mm.h
./virt/kvm/pfn_cache.c
./virt/kvm/vfio.c
./virt/kvm/vfio.h
./virt/lib/
./virt/lib/Kconfig
./virt/lib/Makefile
./virt/lib/irqbypass.c
kheder@kheder:~/usr/src/linux-6.9.3$
```

- I will use default configuration for installation because manually config will take hours just pressing enter and yes / no 😬 😄

⚠️ (This is advice from Amir )

- And now I return to the local terminal because ssh stop working 😬

```
root@kheder:/usr/src/linux-6.9.3# make defconfig
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
#
root@kheder:/usr/src/linux-6.9.3# _
```

- Now let's compile it 💔

After run “ make ” command and waiting about one hour finally

The kernel compiled 😁🥳

```
HOSTCC arch/x86/boot/tools/build
CC arch/x86/boot/compressed/error.o
CPUSTR arch/x86/boot/cpustr.h
CC arch/x86/boot/cpu.o
OBJCOPY arch/x86/boot/compressed/vmlinux.bin
HOSTCC arch/x86/boot/compressed/mkpiggy
CC arch/x86/boot/compressed/cpuflags.o
CC arch/x86/boot/compressed/early_serialconsole.o
CC arch/x86/boot/compressed/kaslr.o
CC arch/x86/boot/compressed/ident_map_64.o
CC arch/x86/boot/compressed/idt_64.o
AS arch/x86/boot/compressed/idt_handlers_64.o
CC arch/x86/boot/compressed/pgtable_64.o
CC arch/x86/boot/compressed/acpi.o
CC arch/x86/boot/compressed/efi.o
AS arch/x86/boot/compressed/efi_mixed.o
CC arch/x86/boot/compressed/misc.o
GZIP arch/x86/boot/compressed/vmlinux.bin.gz
MKPIGGY arch/x86/boot/compressed/piggy.S
AS arch/x86/boot/compressed/piggy.o
LD arch/x86/boot/compressed/vmlinux
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)
oot@kheder:/usr/src/linux-6.9.3#
```

- Now let's install the required modules

```
root@khelder:/usr/src/linux-6.9.3# make modules_install
SYMLINK /lib/modules/6.9.3/build
INSTALL /lib/modules/6.9.3/modules.order
INSTALL /lib/modules/6.9.3/modules.builtin
INSTALL /lib/modules/6.9.3/modules.builtin.modinfo
INSTALL /lib/modules/6.9.3/kernel/fs/efivarfs/efivarfs.ko
INSTALL /lib/modules/6.9.3/kernel/drivers/thermal/intel/x86_pkg_temp_thermal.ko
INSTALL /lib/modules/6.9.3/kernel/net/netfilter/nf_log_syslog.ko
INSTALL /lib/modules/6.9.3/kernel/net/netfilter/xt_mark.ko
INSTALL /lib/modules/6.9.3/kernel/net/netfilter/xt_nat.ko
INSTALL /lib/modules/6.9.3/kernel/net/netfilter/xt_LOG.ko
INSTALL /lib/modules/6.9.3/kernel/net/netfilter/xt_MASQUERADE.ko
INSTALL /lib/modules/6.9.3/kernel/net/netfilter/xt_addrtype.ko
INSTALL /lib/modules/6.9.3/kernel/net/ipv4/netfilter/iptables_nat.ko
DEPMOD /lib/modules/6.9.3
root@khelder:/usr/src/linux-6.9.3#
```

- Then we will install kernel 🐱

```
root@khelder:/usr/src/linux-6.9.3# sudo make install
INSTALL /boot
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 6.9.3 /boot/vmlinuz-6.9.3
update-initramfs: Generating /boot/initrd.img-6.9.3
W: Possible missing firmware /lib/firmware/rtl_nic/rtl8126a-2.fw for built-in driver r8169
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 6.9.3 /boot/vmlinuz-6.9.3
run-parts: executing /etc/kernel/postinst.d/update-notifier 6.9.3 /boot/vmlinuz-6.9.3
run-parts: executing /etc/kernel/postinst.d/xx-update-initrd-links 6.9.3 /boot/vmlinuz-6.9.3
I: /boot/initrd.img.old is now a symlink to initrd.img-5.15.0-112-generic
I: /boot/initrd.img is now a symlink to initrd.img-6.9.3
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.9.3 /boot/vmlinuz-6.9.3
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.9.3
Found initrd image: /boot/initrd.img-6.9.3
Found linux image: /boot/vmlinuz-5.15.0-112-generic
Found initrd image: /boot/initrd.img-5.15.0-112-generic
Found linux image: /boot/vmlinuz-5.15.0-107-generic
Found initrd image: /boot/initrd.img-5.15.0-107-generic
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
root@khelder:/usr/src/linux-6.9.3#
```

- Then just enable the new kernel

sudo update-initramfs -c -k 6.9.3

```
root@khelder:/usr/src/linux-6.9.3# sudo update-initramfs -c -k 6.9.3
update-initramfs: Generating /boot/initrd.img-6.9.3
W: Possible missing firmware /lib/firmware/rtl_nic/rtl8126a-2.fw for built-in driver r8169
root@khelder:/usr/src/linux-6.9.3#
```

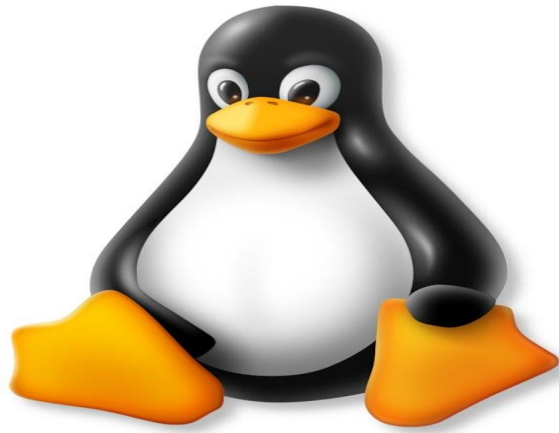
```
root@kheder: /usr/src/linux-6.9.3
root@kheder:/usr/src/linux-6.9.3# sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.9.3
Found initrd image: /boot/initrd.img-6.9.3
Found linux image: /boot/vmlinuz-5.15.0-112-generic
Found initrd image: /boot/initrd.img-5.15.0-112-generic
Found linux image: /boot/vmlinuz-5.15.0-107-generic
Found initrd image: /boot/initrd.img-5.15.0-107-generic
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
root@kheder:/usr/src/linux-6.9.3#
```

- So now let's reboot 😁

After reboot its work but it just a kernel I cannot do so much things 😬

```
(initramfs)
(initramfs)
(initramfs)
(initramfs) ls
dev      bin      etc      lib32    run      usr      proc
root     conf     init     lib64    sbin     var      tmp
kernel   cryptroot lib      libx32   scripts  sys
(initramfs) cd kernel
(initramfs) ls
x86
(initramfs) cd
(initramfs) cd usr
(initramfs) ls
bin      lib      lib32    lib64    libexec  libx32   sbin     share
(initramfs) cd
(initramfs) cd kernel
(initramfs) ls
x86
(initramfs) cd x86/
(initramfs) ls
microcode
(initramfs) cd microcode/
(initramfs) ls
AuthenticAMD.bin  GenuineIntel.bin
(initramfs) _
```





# The Linux Kernel