



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

# گزارش کار آزمایشگاه آزمایشگاه سیستم‌های عامل

گزارش آزمایش شماره ۱  
(آشنایی با سیستم عامل لینوکس)

۲۰

شماره گروه:

ارشیا یوسف‌نیا (۴۰۱۱۰۴۱۵)

گروه:

محمد عارف زارع زاده (۴۰۱۱۰۶۰۱۷)

استاد درس:

دکتر بیگی

تاریخ:

تابستان ۱۴۰۴

## فهرست مطالب

۱	۱	شرح آزمایش
۱	۱,۱	نصب سیستم عامل لینوکس . . . . .
۴	۱,۲	آشنایی با دستورات پایه‌ی لینوکس . . . . .
۱۳	۲	فعالیت‌ها
۱۶	۲,۱	اعمال تغییرات و کامپایل مجدد هسته‌ی سیستم عامل . . . . .
۱۹	۲,۲	فعالیت‌ها (ادامه‌ی کامپایل) . . . . .

## لیست تصاویر

۱	انتخاب ISO Image مربوط به Ubuntu 24.02	۱
۱	انتخاب میزان دسترسی سیستم عامل مجازی به حافظه و پردازندها	۲
۲	ساخت دیسک مجازی و تعیین میزان حجم آن	۳
۲	خلاصه تنظیمات	۴
۳	صفحه boot	۵
۳	انتخاب گزینه minimal	۶
۴	وارد کردن مشخصات کاربر یا کاربرها	۷
۴	نصب سیستم عامل	۸
۵	دستورات pwd و cd و mkdir	۹
۵	محیط ادیتور nano	۱۰
۶	تغییر نام فایل با دستور mv	۱۱
۶	کپی گرفتن از فایل	۱۲
۷	بررسی عملگرهای < و >	۱۳
۸	نوشتن در فایل با کمک دستور cat	۱۴
۸	لیست پردازهای درحال اجرا (بخش اول)	۱۵
۹	لیست پردازهای درحال اجرا (بخش دوم)	۱۶
۹	لیست پردازهای درحال اجرا (بخش سوم)	۱۷
۱۰	لیست پردازهای درحال اجرا (بخش آخر)	۱۸
۱۱	نام پردازهایی که در نامشان حرف a وجود دارد	۱۹
۱۱	کلیه اطلاعات پردازهایی که در آنها حرف a وجود دارد	۲۰
۱۲	بخشی از محتویات دایرکتوری /usr/bin/	۲۱
۱۲	حجم و سایر اطلاعات مربوط به فایلهای /usr/bin/	۲۲
۱۳	نمایش لیست فایلهایی که در آنها عبارات fs یا ld وجود دارد	۲۳
۱۵	پیدا کردن تعداد خطوط در یک فایل متنه	۲۴
۱۶	پیدا کردن تعداد فایلهایی که با حرف A شروع می‌شوند	۲۵
۱۶	نمایش حجم یک فایل	۲۶
۱۷	نمایش نسخه‌ی هسته قبل از کامپایل	۲۷
۱۷	دربیافت کد هسته	۲۸
۱۸	دربیافت ابزارهای لازم برای کامپایل هسته (بخش اول)	۲۹
۱۸	دربیافت ابزارهای لازم برای کامپایل هسته (بخش دوم)	۳۰
۱۹	بازگشایی کدهای هسته و ورود به آن	۳۱
۱۹	محیط make menuconfig	۳۲
۲۰	استفاده از make oldconfig	۳۳
۲۰	کامپایل مجدد هسته	۳۴
۲۱	نصب ماژول‌های هسته	۳۵
۲۱	نصب image هسته	۳۶
۲۲	انتخاب نسخه‌ی جدید هنگام boot	۳۷

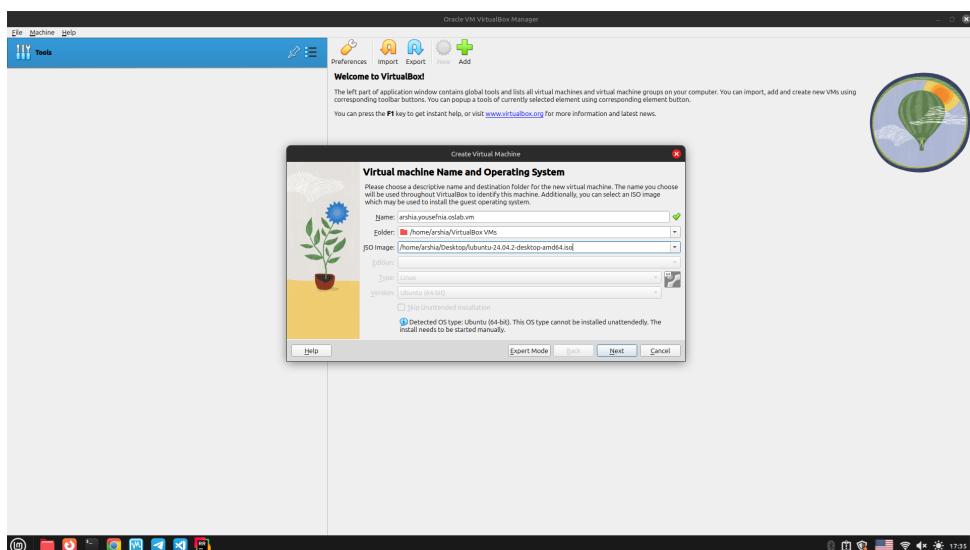


لیست جداول

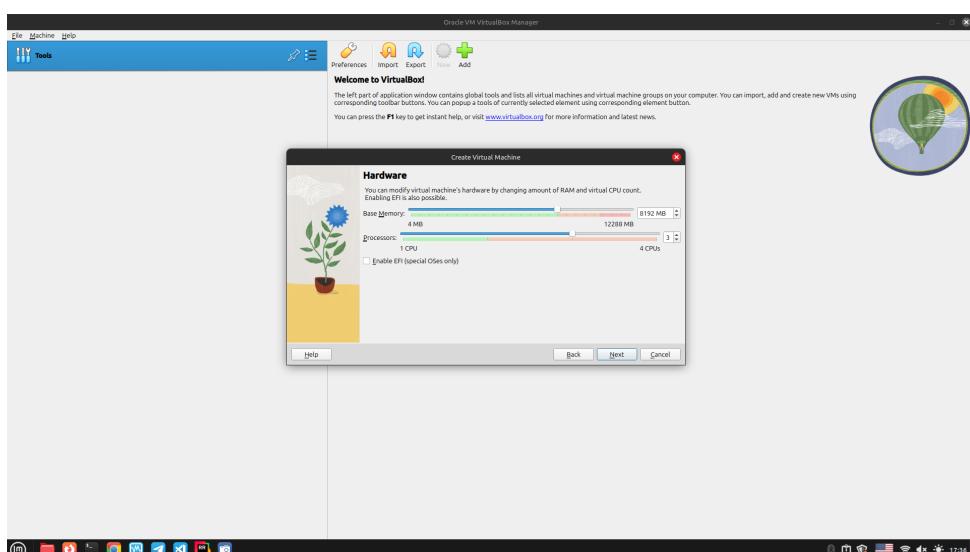
# ۱ شرح آزمایش

## ۱.۱ نصب سیستم عامل لینوکس

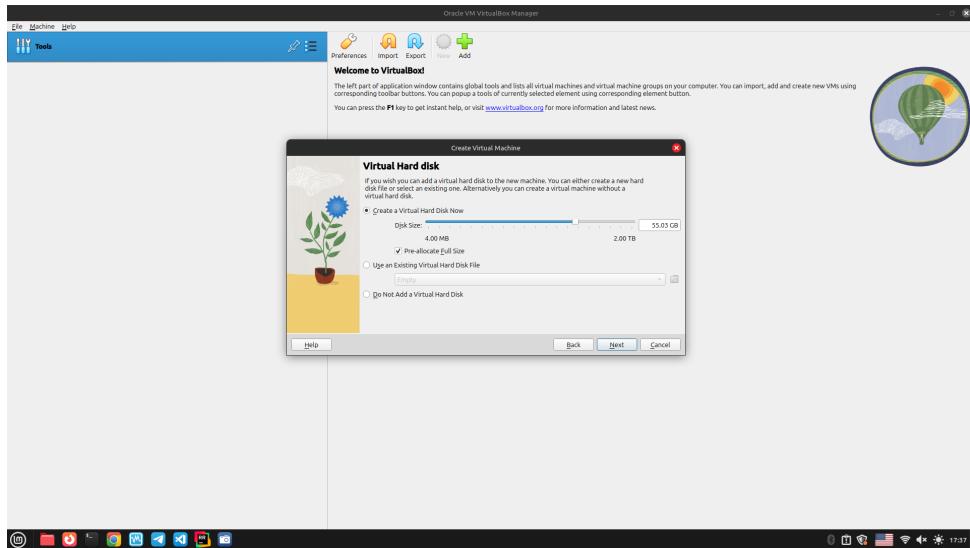
۱. ما از نرم افزار VirtualBox برای مجازی سازی سیستم عامل استفاده می کنیم. به دلیل منسوخ شدن Debian8 مطابق نکته گفته شده در کوئر، از Ubuntu 24.02 استفاده می کنیم. در تصاویر زیر تنظیمات و مراحل راه اندازی را می توانید مشاهده کنید:



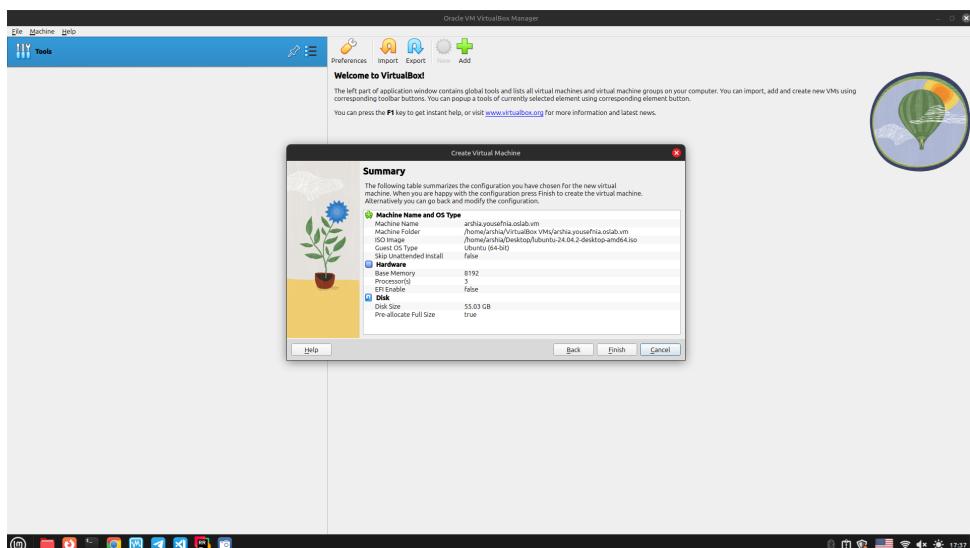
شکل ۱: انتخاب ISO Image مربوط به Ubunto 24.02



شکل ۲: انتخاب میزان دسترسی سیستم عامل مجازی به حافظه و پردازنده ها

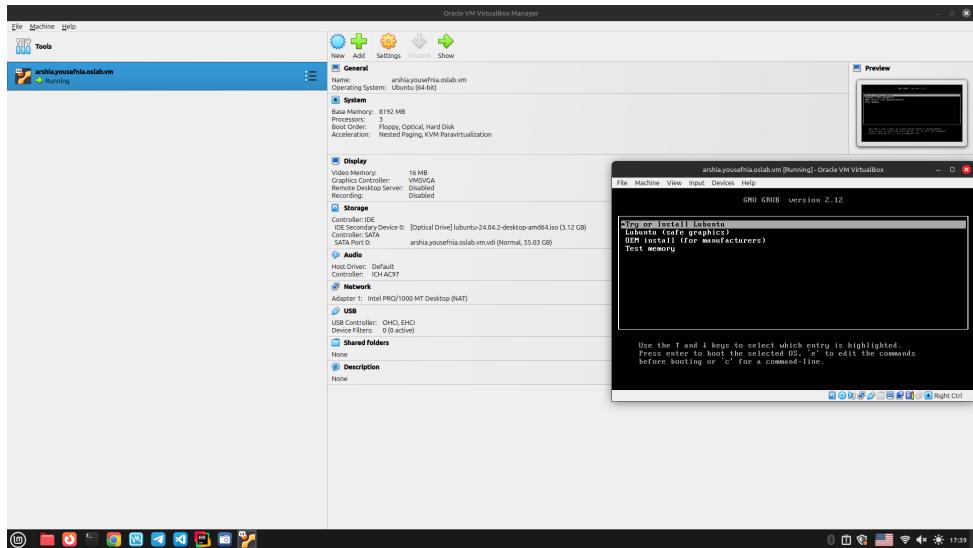


شکل ۳: ساخت دیسک مجازی و تعیین میزان حجم آن

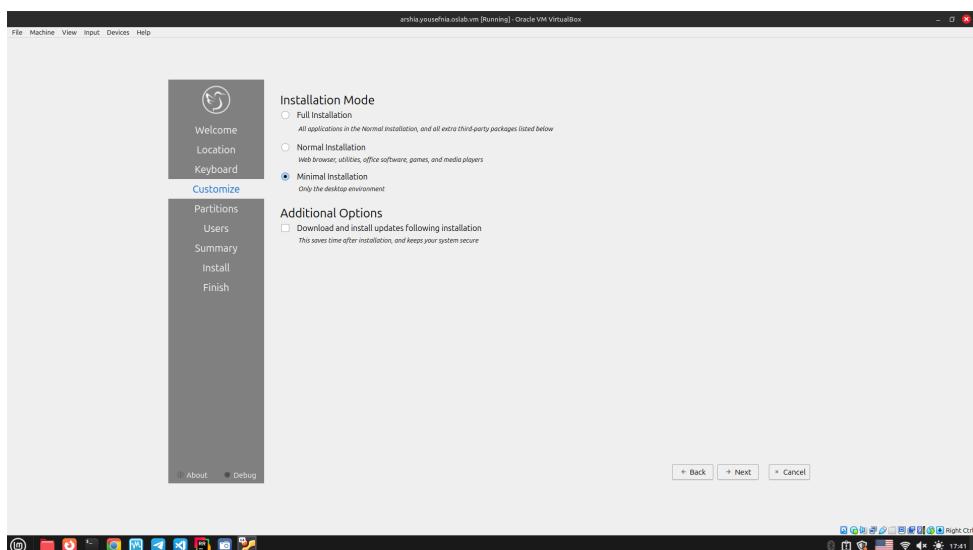


شکل ۴: خلاصهی تنظیمات

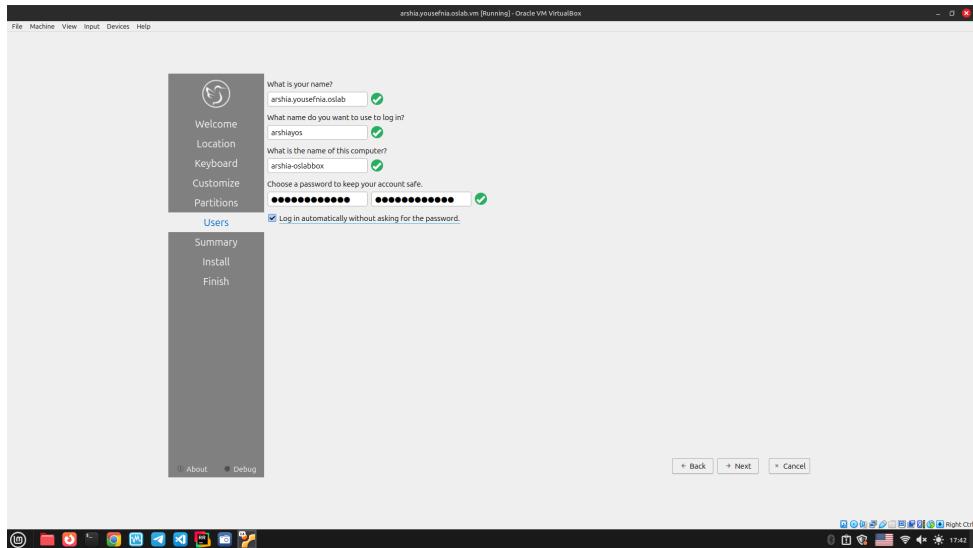
۲. سپس مطابق خواستهی صورت آزمایش، نسخه minimal سیستم عامل را نصب می‌کنیم. هنگام اجرای ماشین مجازی، ابتدا به صفحه boot وارد می‌شویم. پس از انتخاب گزینه نصب سیستم عامل، باید در بخش customization گزینه مینیمال را انتخاب کرده و سپس کاربرهای کاربرهای مورد نظر را ساخته و سیستم عامل را نصب می‌کنیم. در تصاویر زیر، مراحل آن را می‌توان مشاهده کرد.



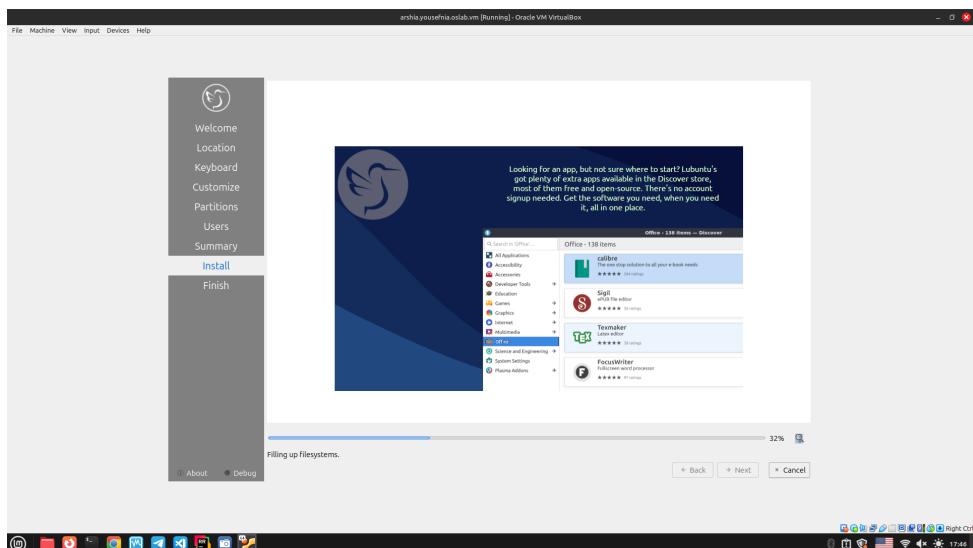
شكل ٥: صفحه boot



شكل ٦: انتخاب گزینه minimal



شکل ۷: وارد کردن مشخصات کاربر یا کاربرها



شکل ۸: نصب سیستم عامل

## ۱،۲ آشنایی با دستورات پایه‌ی لینوکس

۱. با وارد کردن دستور `pwd` در ترمینال، آدرس دایرکتوری فعلی ترمینال چاپ می‌شود. در شکل ۹ می‌توانید خروجی این دستور را مشاهده کنید.
۲. خروجی خواسته‌های این بخش نیز در شکل ۹ قابل مشاهده است.

```

arshiyos@arshia-oslabbox:~/Desktop$ pwd
/home/arshiyos/Desktop
arshiyos@arshia-oslabbox:~/Desktop$ cd /tmp/
arshiyos@arshia-oslabbox:/tmp$ mkdir oslab1
arshiyos@arshia-oslabbox:/tmp$ ls
oslab1
ssdm-auth-12e69dfa-4365-49dc-afa7-c5da9f19a4ee
ssh-SjwVqcTnsk9F
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-ModemManager.service-ny8sNl
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-polkit.service-hgnvP2
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-switcheroo-control.service-jpWQph
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-systemd-logind.service-aIgjoy
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-systemd-resolved.service-xCr64f
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-systemd-timesyncd.service-pseRTR
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-upower.service-Y5lf9p
xauth_jtJuri
arshiyos@arshia-oslabbox:/tmp$ cd oslab1/
arshiyos@arshia-oslabbox:/tmp/oslab1$ 

```

شکل ۹: دستورات pwd و cd و mkdir

۳. ابتدا دستور nano information.txt را می‌زنیم تا فایل خواسته شده ساخته شود. سپس نام و شماره دانشجویی را در آن وارد کرده و با اجرای ctrl+X مانند شکل زیر برنامه از ما می‌پرسد که قصد ذخیره‌ی اطلاعات را داریم یا نه.

```

arshiyos@arshia-oslabbox:~/Desktop$ nano information.txt
name:arshia yousefnia
std:401110415

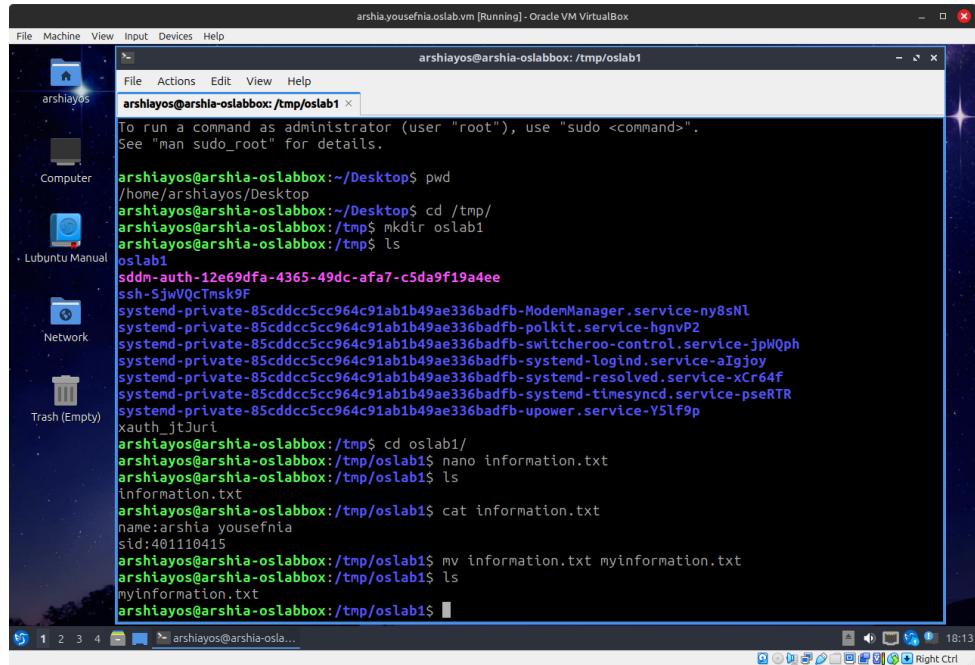
Save modified buffer? [Y/N/C]
Y Yes
N No
C Cancel

```

شکل ۱۰: محیط ادیتور nano

۴. ابتدا با دستور cat نشان می‌دهیم که اطلاعات فایل بخش قبل به درستی ذخیره شده است. سپس با دستور mv آن فایل را کات کرده و با نام myinformation.txt آن را در همان دایرکتوری قرار می‌دهیم. به این

صورت می توانیم نام فایل ها را تغییر دهیم.



```
arshiyos@arshia-oslabbox: /tmp/oslab1
File Actions Edit View Help
arshiyos@arshia-oslabbox: /tmp/oslab1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

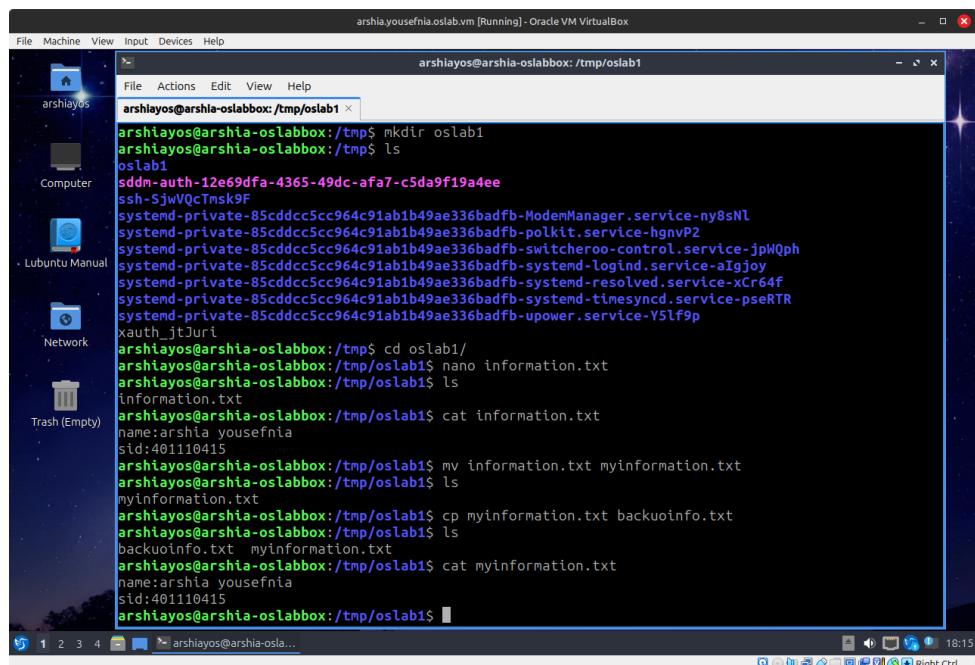
arshiyos@arshia-oslabbox:~/Desktop$ pwd
/home/arshiyos/Desktop
arshiyos@arshia-oslabbox:~/Desktop$ cd /tmp/
arshiyos@arshia-oslabbox:/tmp$ mkdir oslab1
arshiyos@arshia-oslabbox:/tmp$ ls
oslab1
sddm-auth-12e69dfa-4365-49dc-afa7-c5da9f19a4ee
ssh-SjwVQcTnsk9F
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-ModemManager.service-ny8sNl
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-polkit.service-hgnvP2
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-switcheroo-control.service-jpWQph
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-systemd-logind.service-aIgjoy
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-systemd-resolved.service-xCr64f
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-systemd-timesyncd.service-pseRTR
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-upower.service-Y5lf9p
xauth_jtJuri
arshiyos@arshia-oslabbox:/tmp$ cd oslab1/
arshiyos@arshia-oslabbox:/tmp/oslab1$ nano information.txt
arshiyos@arshia-oslabbox:/tmp/oslab1$ ls
information.txt
arshiyos@arshia-oslabbox:/tmp/oslab1$ cat information.txt
name:arshia yousefnia
sid:401110415
arshiyos@arshia-oslabbox:/tmp/oslab1$ mv information.txt myinformation.txt
myinformation.txt
arshiyos@arshia-oslabbox:/tmp/oslab1$
```

شکل ۱۱ : تغییر نام فایل با دستور mv

## ۵. با دستور cp و با فرمت

cp <src-address> <dst-address>

می توانیم از فایل ها کپی بگیریم.

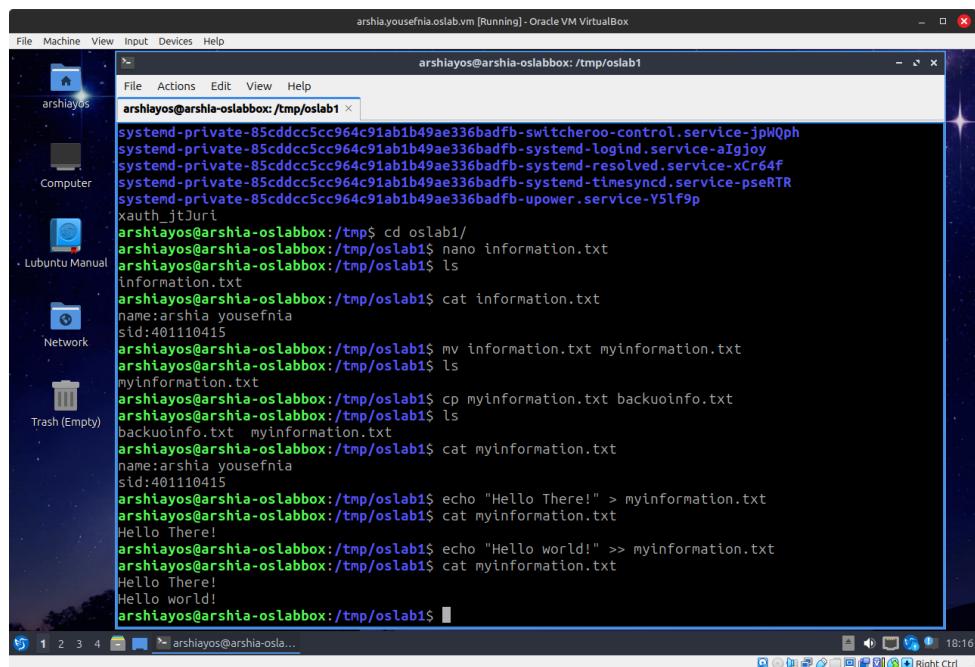


```
arshiyos@arshia-oslabbox: /tmp/oslab1
File Actions Edit View Help
arshiyos@arshia-oslabbox: /tmp/oslab1
arshiyos@arshia-oslabbox: /tmp$ mkdir oslab1
arshiyos@arshia-oslabbox: /tmp$ ls
oslab1
sddm-auth-12e69dfa-4365-49dc-afa7-c5da9f19a4ee
ssh-SjwVQcTnsk9F
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-ModemManager.service-ny8sNl
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-polkit.service-hgnvP2
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-switcheroo-control.service-jpWQph
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-systemd-logind.service-aIgjoy
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-systemd-resolved.service-xCr64f
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-systemd-timesyncd.service-pseRTR
systemd-private-85cdcc5cc964c91ab1b49ae336badfb-upower.service-Y5lf9p
xauth_jtJuri
arshiyos@arshia-oslabbox:/tmp$ cd oslab1/
arshiyos@arshia-oslabbox:/tmp/oslab1$ nano information.txt
arshiyos@arshia-oslabbox:/tmp/oslab1$ ls
information.txt
arshiyos@arshia-oslabbox:/tmp/oslab1$ cat information.txt
name:arshia yousefnia
sid:401110415
arshiyos@arshia-oslabbox:/tmp/oslab1$ mv information.txt myinformation.txt
myinformation.txt
arshiyos@arshia-oslabbox:/tmp/oslab1$ cp myinformation.txt backuoinfo.txt
arshiyos@arshia-oslabbox:/tmp/oslab1$ ls
backuoinfo.txt myinformation.txt
arshiyos@arshia-oslabbox:/tmp/oslab1$ cat myinformation.txt
name:arshia yousefnia
sid:401110415
arshiyos@arshia-oslabbox:/tmp/oslab1$
```

شکل ۱۲ : کپی گرفتن از فایل

۶. همانطور که از شکل ۱۲ می‌توان مشاهده کرد، محتويات فایل کبی شده و فایل اصلی یکسان است.

۷. در تصویر زیر، می‌توان دو دستور گفته شده را مشاهده کرده و خروجی آنها را دید. همانطور که از تصویر قابل مشاهده است، استفاده از عملگر < خروجی دستور سمت چپش را در فایل سمت راست می‌ریزد، و طی آن محتويات قبلی آن فایل از بین می‌رود. اما استفاده از عملگر >> محتويات خروجی دستور را به آن فایل می‌کند. یعنی آن را به انتهای محتويات آن فایل اضافه می‌کند.



```
arshayos@arshia-oslabox: /tmp/oslab1
File Actions Edit View Help
arshayos@arshia-oslabox: /tmp/oslab1 ~
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-switcheroo-control.service-jpWQph
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-systemd-logind.service-aIgjoy
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-systemd-resolved.service-xCr64f
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-systemd-timesyncd.service-pseRTR
systemd-private-85cddcc5cc964c91ab1b49ae336badfb-upower.service-Y5lf9p
xauth_jtJuri
arshayos@arshia-oslabox:/tmp$ cd oslab1/
arshayos@arshia-oslabox:/tmp/oslab1$ nano information.txt
arshayos@arshia-oslabox:/tmp/oslab1$ ls
information.txt
arshayos@arshia-oslabox:/tmp/oslab1$ cat information.txt
name:arshia yousefnia
std:401110415
arshayos@arshia-oslabox:/tmp/oslab1$ mv information.txt myinformation.txt
arshayos@arshia-oslabox:/tmp/oslab1$ ls
myinformation.txt
arshayos@arshia-oslabox:/tmp/oslab1$ cp myinformation.txt backuoinfo.txt
arshayos@arshia-oslabox:/tmp/oslab1$ ls
backuoinfo.txt myinformation.txt
arshayos@arshia-oslabox:/tmp/oslab1$ cat myinformation.txt
name:arshia yousefnia
std:401110415
arshayos@arshia-oslabox:/tmp/oslab1$ echo "Hello There!" > myinformation.txt
arshayos@arshia-oslabox:/tmp/oslab1$ cat myinformation.txt
Hello There!
arshayos@arshia-oslabox:/tmp/oslab1$ echo "Hello world!" >> myinformation.txt
arshayos@arshia-oslabox:/tmp/oslab1$ cat myinformation.txt
Hello There!
Hello world!
arshayos@arshia-oslabox:/tmp/oslab1$
```

شکل ۱۳: بررسی عملگرهای < و >>

۸. می‌توان با دستور `cat` و با کمک عملگر <, بدون استفاده از ادیتور، فایل جدید با محتوای دلخواه (یا فایل قدیم با محتوای جدید) ساخت. وقتی دستور را به فرمت

`cat <filename>`

وارد می‌کنیم، می‌توان هر محتوایی که خواستیم بنویسیم و تا وقتی که از `ctrl+C` استفاده نکرده‌ایم، هر چیزی که در ترمینال می‌نویسیم (حتی `\n`) در فایل قرار می‌گیرد.

```

arshiyos@arshia-oslabbox:/tmp/oslab1$ cat > testfile.txt
this is line 1
this is line 2
this is line 3
this is line 4
^C
arshiyos@arshia-oslabbox:/tmp/oslab1$ cat testfile.txt
this is line 1
this is line 2
this is line 3
this is line 4
arshiyos@arshia-oslabbox:/tmp/oslab1$ 

```

شکل ۱۴: نوشتن در فایل با کمک دستور cat

.۹. با اجرای دستور ps aux می‌توانیم لیست کل پردازه‌های در حال اجرا را مشاهده کنیم. در شکل‌های ۱۵ تا ۱۸ می‌توان این پردازه‌ها را دید.

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1318	0.0	0.0	0	0	?	I	18:19	0:00	[kworker/u14:1-events_power_ef]
root	1322	0.0	0.0	13616	4572	pts/0	R+	18:20	0:00	ps aux
root	1	0.1	0.1	22432	13304	?	Ss	18:06	0:01	/sbin/init splash
root	2	0.0	0.0	0	0	?	S	18:06	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	18:06	0:00	[pool_workqueue_release]
root	4	0.0	0.0	0	0	?	I<	18:06	0:00	[kworker/R-rcu_gp]
root	5	0.0	0.0	0	0	?	I<	18:06	0:00	[kworker/R-sync_wq]
root	6	0.0	0.0	0	0	?	I<	18:06	0:00	[kworker/R-slub_flushwq]
root	7	0.0	0.0	0	0	?	I<	18:06	0:00	[kworker/R-netsns]
root	8	0.0	0.0	0	0	?	I	18:06	0:00	[kworker/0-events]
root	10	0.0	0.0	0	0	?	I<	18:06	0:00	[kworker/0:0-kblockd]
root	11	0.0	0.0	0	0	?	I	18:06	0:00	[kworker/u12:0-ipv6_addrconf]
root	12	0.0	0.0	0	0	?	I<	18:06	0:00	[kworker/R-mm_percpu_wq]
root	13	0.0	0.0	0	0	?	I	18:06	0:00	[rcu_tasks_kthread]
root	14	0.0	0.0	0	0	?	I	18:06	0:00	[rcu_tasks_rude_kthread]
root	15	0.0	0.0	0	0	?	I	18:06	0:00	[rcu_tasks_trace_kthread]
root	16	0.0	0.0	0	0	?	S	18:06	0:00	[ksoftirqd/0]
root	17	0.0	0.0	0	0	?	I	18:06	0:00	[rcu_preempt]
root	18	0.0	0.0	0	0	?	S	18:06	0:00	[rcu_exp_par_gp_kthread_worker]
root	19	0.0	0.0	0	0	?	S	18:06	0:00	[rcu_exp_gp_kthread_worker]
root	20	0.0	0.0	0	0	?	S	18:06	0:00	[migration/0]
root	21	0.0	0.0	0	0	?	S	18:06	0:00	[idle_inject/0]
root	22	0.0	0.0	0	0	?	S	18:06	0:00	[cpuhp/0]
root	23	0.0	0.0	0	0	?	S	18:06	0:00	[cpuhp/1]
root	24	0.0	0.0	0	0	?	S	18:06	0:00	[idle_inject/1]
root	25	0.0	0.0	0	0	?	S	18:06	0:00	[migration/1]
root	26	0.0	0.0	0	0	?	S	18:06	0:00	[kssoftirqd/1]
root	28	0.0	0.0	0	0	?	I<	18:06	0:00	[kworker/1:0H-events_highpri]

شکل ۱۵: لیست پردازه‌های درحال اجرا (بخش اول)

```

arshiyos@arshia-oslabbbox: /tmp/oslab1
root      29  0.0  0.0   0  0 ?      S  18:06  0:00 [cpuhp/2]
root      30  0.0  0.0   0  0 ?      S  18:06  0:00 [idle_inject/2]
root      31  0.0  0.0   0  0 ?      S  18:06  0:00 [migration/2]
root      32  0.0  0.0   0  0 ?      S  18:06  0:00 [ksoftirqd/2]
root      34  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/2:0H-events_highpri]
root      36  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u14:0-events_unbound]
root      37  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u15:0-flush-8:0]
root      38  0.0  0.0   0  0 ?      S  18:06  0:00 [kdevtmpfs]
root      39  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-inet_frag_wq]
root      40  0.0  0.0   0  0 ?      S  18:06  0:00 [kaudittd]
root      41  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u13:1-events_unbound]
root      42  0.0  0.0   0  0 ?      S  18:06  0:00 [khungtaskd]
root      43  0.0  0.0   0  0 ?      S  18:06  0:00 [oom_reaper]
root      45  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-writeback]
root      46  0.0  0.0   0  0 ?      S  18:06  0:00 [kcompactd0]
root      47  0.0  0.0   0  0 ?      SN 18:06  0:00 [ksmd]
root      48  0.0  0.0   0  0 ?      SN 18:06  0:00 [khugepaged]
root      49  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-kintegrityd]
root      50  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-kblockd]
root      51  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-blkcg_punt_bio]
root      52  0.0  0.0   0  0 ?      S  18:06  0:00 [irq/9-acpi]
root      53  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/I:1-events]
root      54  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-tpm_dev_wq]
root      55  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-ata_sff]
root      56  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-md]
root      57  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-md_bitmap]
root      58  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-edac_poller]
root      59  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-devfreq_wq]
root      60  0.0  0.0   0  0 ?      S  18:06  0:00 [watchdogd]
root      61  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/2:1-events]

```

شکل ۱۶: لیست پردازهای درحال اجرا (بخش دوم)

```

arshiyos@arshia-oslabbbox: /tmp/oslab1
root      62  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/1:1H-kblockd]
root      63  0.0  0.0   0  0 ?      S  18:06  0:00 [kswapd0]
root      64  0.0  0.0   0  0 ?      S  18:06  0:00 [encryptfs-kthread]
root      65  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-kthrotld]
root      66  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-acpi_thermal_pm]
root      68  0.0  0.0   0  0 ?      S  18:06  0:00 [scsi_eh_0]
root      69  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-scscsi_tmfs_0]
root      70  0.0  0.0   0  0 ?      S  18:06  0:00 [scsi_eh_1]
root      71  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-scscsi_tmfs_1]
root      74  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u14:5-events_unbound]
root      75  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-mld]
root      76  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-ipv6_addrconf]
root      77  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u12:1-ipv6_addrconf]
root      85  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-kstrt]
root      87  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u16:0]
root      88  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u17:0]
root      89  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u18:0]
root      90  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u19:0]
root      92  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/1:2-cgroup_destroy]
root      96  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-cryptd]
root     106  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-charger_manager]
root     156  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/2:1H-kblockd]
root     157  0.0  0.0   0  0 ?      S  18:06  0:00 [scsi_eh_2]
root     158  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-scscsi_tmfs_2]
root     160  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u13:2-events_unbound]
root     170  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u12:1H-kblockd]
root     178  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/u15:2-events_unbound]
root     227  0.0  0.0   0  0 ?      S  18:06  0:00 [jbd2/sda1-8]
root     228  0.0  0.0   0  0 ?      I< 18:06  0:00 [kworker/R-ext4-rsv-conversion]
root    274  0.0  0.1 50444 16172 ?  S<s 18:06  0:00 [/usr/lib/lib/systemd/systemd-journ]

```

شکل ۱۷: لیست پردازهای درحال اجرا (بخش سوم)

```

arshiyos@arshya-oslabox: /tmp/oslab1
arshiyay+ 1038 0.0 0.5 340228 41768 ? Sl 18:07 0:00 /usr/bin/lxqt-globalkeysd
arshiyay+ 1039 0.0 1.2 455348 100924 ? Sl 18:07 0:00 /usr/bin/lxqt-notificationd
arshiyay+ 1040 0.0 1.4 763244 118832 ? Sl 18:07 0:00 /usr/bin/lxqt-panel
arshiyay+ 1041 0.0 0.5 487728 43372 ? Sl 18:07 0:00 /usr/bin/lxqt-policykit-agent
arshiyay+ 1042 0.0 0.6 290168 56848 ? Sl 18:07 0:00 /usr/bin/lxqt-runner
arshiyay+ 1044 0.0 0.0 4644 3560 ? S 18:07 0:00 /usr/bin/xscreensaver -no-splash
arshiyay+ 1050 0.0 0.0 9420 4912 ? S 18:07 0:00 /usr/bin/dbus-daemon --config=
arshiyay+ 1054 0.0 0.4 63384 36372 ? S 18:07 0:00 /usr/bin/python3 /usr/share/sy
arshiyay+ 1076 0.0 0.0 7848 4048 ? S 18:07 0:00 xscreensaver-systemd
arshiyay+ 1082 0.0 0.0 236016 8848 ? Sl 18:07 0:00 /usr/libexec/at-spi2-registryd
arshiyay+ 1094 0.0 0.0 314160 7856 ? Ssl 18:07 0:00 /usr/libexec/gvfsd
root 1095 0.0 0.0 0 0 ? I 18:07 0:00 [kworker/u15:4-events_power_ef
arshiyay+ 1101 0.0 0.0 459712 6908 ? Sl 18:07 0:00 /usr/libexec/gvd-fuse /run/u
arshiyay+ 1102 0.0 0.1 535712 8568 ? Sl 18:07 0:00 /usr/libexec/gvfsd-trash --spa
arshiyay+ 1118 0.0 0.1 389128 9776 ? Ssl 18:07 0:00 /usr/libexec/gvfs-udisks2-volu
arshiyay+ 1124 0.0 0.0 389376 7792 ? Ssl 18:07 0:00 /usr/libexec/gvfs-afc-volume-m
arshiyay+ 1131 0.0 0.0 309792 6536 ? Ssl 18:07 0:00 /usr/libexec/gvfs-ntp-volume-m
arshiyay+ 1137 0.0 0.0 310760 6964 ? Ssl 18:07 0:00 /usr/libexec/gvfs-qphoto2-volu
arshiyay+ 1143 0.0 0.0 309772 6312 ? Ssl 18:07 0:00 /usr/libexec/gvfs-goa-volume-m
arshiyay+ 1169 0.0 0.0 236304 6292 ? Ssl 18:07 0:00 /usr/libexec/gvfsd-metadata
root 1184 0.0 0.1 316652 8956 ? Ssl 18:07 0:00 /usr/libexec/upowerd
arshiyay+ 1190 0.0 0.5 357808 46384 ? Sl 18:07 0:00 /usr/bin/lxqt-powermanagement
arshiyay+ 1192 0.0 0.5 263156 41788 ? Sl 18:07 0:00 /usr/bin/nm-tray
arshiyay+ 1208 0.0 0.1 49072 8200 ? Ss 18:07 0:00 /usr/libexec/blueuetooth/obexd
arshiyay+ 1238 0.3 1.3 455784 106452 ? Sl 18:08 0:02 qterminal
arshiyay+ 1241 0.0 0.0 11252 5508 pts/0 Ss 18:08 0:00 /bin/bash
root 1312 0.0 0.0 0 0 ? I 18:16 0:00 [kworker/u15:1-flush-8:0]
root 1318 0.0 0.0 0 0 ? I 18:19 0:00 [kworker/u14:1-events_unbound]
arshiyay+ 1323 0.0 0.0 13616 4588 pts/0 R+ 18:20 0:00 ps aux

```

شکل ۱۸: لیست پردازه‌های درحال اجرا (بخش آخر)

۱۰. با دستور `ps` و استفاده از آپشن `-e` می‌توان تمام پردازه‌ها را مشاهده کرد. همچنین با آپشن `-o` می‌توان نام ستون‌هایی که می‌خواهیم در خروجی بیایند را مشخص کرد. برای مثال `comm` برای نام پردازه به کار می‌رود (به طور دقیق‌تر نام دستوری که مربوط به آن پردازه است).

بنابراین با دستور `ps -eo comm` می‌توان نام تمام پردازه‌ها را اجرا کرد. همچنین با استفاده از عملگر | که خروجی دستور سمت چپ را به عنوان ورودی به دستور سمت راست می‌دهد، می‌توان روی لیست تمام پردازه‌ها، دستور `grep` را اجرا کرد و با کمک "a" `grep "a"` می‌توان نام تمام پردازه‌هایی که در آنها حرف a وجود دارد را مشاهده کرد. در شکل ۱۹ فقط نام آن پردازه‌ها طبق توضیحات گفته شده آورده شده است، و در شکل ۲۰ کلیه اطلاعات آورده شده‌اند.

```

arshiyos@arshia-oslabbbox:/tmp/oslab1$ ps -eo comm | grep "a"
kthreadd
pool_workqueue_release
kworker/u12:0-ipv6_addrconf
rcu_tasks_kthread
rcu_tasks_rude_kthread
rcu_tasks_trace_kthread
rcu_exp_par_gp_kthread_worker/0
rcu_exp_gp_kthread_worker
migration/0
migration/1
migration/2
kworker/R-inet_frag_wq
kauditd
khungtaskd
oom_reaper
kworker/R-writeback
kcompactd0
khugepaged
irq/9-acpi
kworker/R-ata_sff
kworker/R-md_bitmap
kworker/R-edac-polter
watchdogd
kswapd0
ecryptfs-kthread
kworker/R-acpi_thermal_pm
kworker/R-ipv6_addrconf
kworker/u12:1-ipv6_addrconf
kworker/R-charger_manager

```

شکل ۱۹: نام پردازه‌هایی که در نامشان حرف a وجود دارد

```

arshiyos@arshia-oslabbbox:/tmp/oslab1$ ps aux | grep "a"
root      1  0.0  0.1 22432 13304 ?        Ss   18:06  0:01 /sbin/init splash
root      2  0.0  0.0      0  0 ?        S     18:06  0:00 [kthreadd]
root      3  0.0  0.0      0  0 ?        S     18:06  0:00 [pool_workqueue_release]
root     11  0.0  0.0      0  0 ?        I     18:06  0:00 [kworker/u12:0-ipv6_addrconf]
root     13  0.0  0.0      0  0 ?        I     18:06  0:00 [rcu_tasks_kthread]
root     14  0.0  0.0      0  0 ?        I     18:06  0:00 [rcu_tasks_rude_kthread]
root     15  0.0  0.0      0  0 ?        I     18:06  0:00 [rcu_tasks_trace_kthread]
root     18  0.0  0.0      0  0 ?        S     18:06  0:00 [rcu_exp_par_gp_kthread_worker]
/
root    19  0.0  0.0      0  0 ?        S     18:06  0:00 [rcu_exp_gp_kthread_worker]
root    20  0.0  0.0      0  0 ?        S     18:06  0:00 [migration/0]
root    25  0.0  0.0      0  0 ?        S     18:06  0:00 [migration/1]
root    31  0.0  0.0      0  0 ?        S     18:06  0:00 [migration/2]
root    39  0.0  0.0      0  0 ?        I<   18:06  0:00 [kworker/R-inet_frag_wq]
root    40  0.0  0.0      0  0 ?        S     18:06  0:00 [kauditd]
root    42  0.0  0.0      0  0 ?        S     18:06  0:00 [khungtaskd]
root    43  0.0  0.0      0  0 ?        S     18:06  0:00 [oom_reaper]
root    45  0.0  0.0      0  0 ?        I<   18:06  0:00 [kworker/R-writeback]
root    46  0.0  0.0      0  0 ?        S     18:06  0:00 [kcompactd0]
root    48  0.0  0.0      0  0 ?        SN    18:06  0:00 [khugepaged]
root    52  0.0  0.0      0  0 ?        S     18:06  0:00 [irq/9-acpi]
root    55  0.0  0.0      0  0 ?        I<   18:06  0:00 [kworker/R-ata_sff]
root    57  0.0  0.0      0  0 ?        I<   18:06  0:00 [kworker/R-md_bitmap]
root    58  0.0  0.0      0  0 ?        I<   18:06  0:00 [kworker/R-edac-polter]
root    60  0.0  0.0      0  0 ?        S     18:06  0:00 [watchdogd]
root    63  0.0  0.0      0  0 ?        S     18:06  0:00 [kswapd0]
root    64  0.0  0.0      0  0 ?        S     18:06  0:00 [ecryptfs-kthread]
root    66  0.0  0.0      0  0 ?        I<   18:06  0:00 [kworker/R-acpi_thermal_pm]
root    76  0.0  0.0      0  0 ?        I<   18:06  0:00 [kworker/R-ipv6_addrconf]

```

شکل ۲۰: کلیه اطلاعات پردازه‌هایی که در آنها حرف a وجود دارد

۱۱. خروجی دستور ls در این بخش را در شکل زیر می‌توانید مشاهده کنید.

```

arshayos@arshia-oslabbox:~$ cd /usr/bin/
arshayos@arshia-oslabbox:/usr/bin$ ls
'!' pamstretch
411toppm pamstretch-gen
7z pamsumm
7za pamsummcol
7zr pamtable
aa-enabled pamthreshold
aa-exec pamtilt
aa-features-abi pamtoavs
aconnect pamtodjvurle
acpidbg pantofits
add-apt-repository pantogif
adpart pantohdiff
airscan-discover pantohntbl
alsabat pamtojpeg2k
alsaloop pamtopdfont
alsamixer pamtoctaveimg
alsatplg pantopan
alsaucm pantopdbmg
amidi pantopfm
amixer pantopng
animate pantopnm
animate-im6 pantoqoi
animate-im6.q16 pantosrf
anytopnm pantosvg
aplay pamtotga
aplaymidi pantotiff
aport-bug pantouil
aport-cli pantowinicon

```

شکل ۲۱: بخشی از محتويات دایرکتوری /usr/bin/

۱۲. آپشن ۱- اطلاعات اضافه‌تر درمورد فایل‌ها مانند مالک فایل، سطوح دسترسی فایل، حجم فایل و ... را نشان می‌دهد. همچنین آپشن h- حجم فایل‌ها را به فرمتی قابل فهم‌تر (برای مثال برحسب کیلوبایت یا مگابایت) بیان می‌کند.

همچنین با دستور head برای اینکه صفحه شلوغ نشود، فقط ۲۰ فایل اول را نشان می‌دهیم.

```

arshayos@arshia-oslabbox:~$ ls -lh | head -n 20
total 156M
-rwxr-xr-x 1 root root      55K Apr  5 2024 [REDACTED]
-rwxr-xr-x 1 root root     15K Mar 31 2024 411toppm
-rwxr-xr-x 1 root root     38K Apr 11 2024 7z
-rwxr-xr-x 1 root root     39K Apr 11 2024 7za
-rwxr-xr-x 1 root root     39K Apr 11 2024 7zr
-rwxr-xr-x 1 root root    19K Jul 18 2024 aa-enabled
-rwxr-xr-x 1 root root    19K Jul 18 2024 aa-exec
-rwxr-xr-x 1 root root    19K Jul 18 2024 aa-features-abi
-rwxr-xr-x 1 root root   23K Apr  7 2024 aconnect
-rwxr-xr-x 1 root root   1.6K Jan 17 15:35 acpidbg
-rwxr-xr-x 1 root root   17K Aug 15 2024 add-apt-repository
-rwxr-xr-x 1 root root   15K Dec  5 2024 adpart
-rwxr-xr-x 1 root root  155K Apr  8 2024 airsanc-discover
-rwxr-xr-x 1 root root   47K Apr  7 2024 alsabat
-rwxr-xr-x 1 root root   84K Apr  7 2024 alsaloop
-rwxr-xr-x 1 root root   81K Apr  7 2024 alsamixer
-rwxr-xr-x 1 root root   75K Apr  7 2024 alsatplg
-rwxr-xr-x 1 root root   31K Apr  7 2024 alsaucm
-rwxr-xr-x 1 root root   31K Apr  7 2024 amidi
arshayos@arshia-oslabbox:/usr/bin$ 

```

شکل ۲۲: حجم و سایر اطلاعات مربوط به فایل‌های /usr/bin/

۱۳. در بخش‌های قبل، یک نمونه از کارکرد دستور grep را دیدیم. برای اینکه بتوان عبارات جستجوی پیچیده‌تر

را به آن داد، می‌توان از آپشن E- استفاده کرد که با عبارت جستجو، مانند یک عبارت رجکس رفتار می‌کند. می‌دانیم در رجکس | معادل or است، پس مطابق شکل زیر، می‌توان به خواسته‌ی صورت آزمایش رسید.

```
arshiyos@arshia-oslabbox:/usr/bin$ ls -lh | grep -E "fs|ld"
arshiyos@arshia-oslabbox:/usr/bin$ ls -lh | grep -E "fs|ld"
-rwxr-xr-x 1 root root 1002K Mar 31 2024 btrfs
-rw-r--r-- 1 root root 5 Mar 31 2024 btrfsck -> btrfs
-rwxr-xr-x 1 root root 577K Mar 31 2024 btrfs-convert
-rwxr-xr-x 1 root root 521K Mar 31 2024 btrfs-find-root
-rwxr-xr-x 1 root root 553K Mar 31 2024 btrfs-image
-rwxr-xr-x 1 root root 521K Mar 31 2024 btrfs-map-logical
-rwxr-xr-x 1 root root 517K Mar 31 2024 btrfs-select-super
-rwxr-xr-x 1 root root 545K Mar 31 2024 btrfs-tune
-rwxr-xr-x 1 root root 27K Jan 10 2024 cifsiostat
-rwxr-xr-x 1 root root 35K Apr 5 2024 fold
-rwxr-xr-x 1 root root 15K Mar 31 2024 fstopgm
-rwxr-xr-x 1 root root 15K Mar 31 2024 gouldtoppm
-rwxr-xr-x 1 root root 859K Sep 11 2024 grub-fstab
lrwxrwxrwx 1 root root 34 Sep 11 2024 grub-ntldr-img -> ../../lib/grub/i386-pc/grub-ntldr-img
-rwxr-xr-x 1 root root 75K Apr 4 2024 gtk4-builder-tool
-rwxr-xr-x 1 root root 35K Aug 28 2024 gtk-builder-tool
-rwxr-xr-x 1 root root 31K Apr 11 2024 hplibdecode
-rwxr-xr-x 1 root root 19K Mar 31 2024 kbuildsycoca5
-rwxr-xr-x 1 root root 5.3K Jan 28 20:37 ld
lrwxrwxrwx 1 root root 29 Jan 28 20:37 ld.so -> ../../lib64/ld-linux-x86-64.so.2
-rwxr-xr-x 1 root root 119K Apr 8 2024 lowntfs-3g
-rwxr-xr-x 1 root root 735 Sep 20 2024 lsinitramfs
lrwxrwxrwx 1 root root 11 Mar 14 2024 mkisofs -> genisoimage
-rwxr-xr-x 1 root root 160K Apr 8 2024 ntfs-3g
-rwxr-xr-x 1 root root 15K Apr 8 2024 ntfs-3g.probe
-rwxr-xr-x 1 root root 27K Apr 8 2024 ntfsccat
-rwxr-xr-x 1 root root 39K Apr 8 2024 ntfscluster
-rwxr-xr-x 1 root root 31K Apr 8 2024 ntfsncmp
-rwxr-xr-x 1 root root 43K Apr 8 2024 ntfsdecrypt
```

شکل ۲۳: نمایش لیست فایل‌هایی که در آنها عبارات fs یا ld وجود دارد

## ۲ فعالیت‌ها

- دستور cut: [۱] این دستور برای اینکه محتویات یک فایل را براساس یک کاراکتر یا عبارت در هر خط تقسیم کند و بخش‌های خاصی از آن را نمایش دهد. به بیان دیگر، این دستور فایل گفته شده را خط به خط بررسی کرده، و بسته به اینکه کدام آپشن‌ها استفاده می‌شوند، یا آن را با توجه به یک کاراکتر به چندین عبارت تقسیم کرده و عباراتی که کاربر می‌گوید را نشان می‌دهد، یا صرفاً بایت‌ها یا کاراکترهایی که کاربر می‌گوید را در هر خط نشان می‌دهد. قالب کلی آن به این صورت است:

`cut [options] [file]`

از آپشن‌های پرکاربر آن می‌توان به -c- یا -b- و همچنین -d- و -f- اشاره کرد. دو آپشن آخر با هم می‌توانند در یک دستور قرار بگیرند، ولی سایر ترکیب آپشن‌ها نمی‌توانند.

اگر از آپشن -c- یا -b- استفاده شده باشد، کاربر می‌تواند به صورت یک لیست از چند بازه، شماره کاراکترها یا بایت‌هایی که می‌خواهد از هر خط نمایش داده شوند را تغیین کند.

اگر از آپشن -d- که جلوی آن باید حتماً یک کاراکتر قرار بگیرد استفاده شود، هر خط از فایل توسط این کاراکتر به چند عبارت تقسیم شده، و شماره عباراتی که با آپشن -f- مشخص شده نمایش داده می‌شود. برای مثال اگر یک فایل csv که اعضاً هر خانه توسط کاما از هم جدا شده اند داشته باشیم، و بخواهیم ستون اول و سوم آن را مشاهده کنیم، می‌توانیم از این دستور استفاده کنیم:

`cut -d',' -f1,3 data.csv`

- دستور find: این دستور برای جستجو بین فایل‌ها به کار می‌رود. قالب کلی آن به صورت

## find [path] [expression]

است. بخش path مربوط به دایرکتوری جستجو است، و expression جستجو بر اساس ویژگی های فایل است. از جمله آپشن هایی که در این بخش می توانند قرار بگیرند، می توان به name- اشاره کرد که براساس نام فایل جستجو را انجام می دهد، و قابلیت پشتیبانی از wildcard را دارد، size- که براساس سایز فایل جستجو را انجام می دهد (برای مثال فایل های بزرگتر از 100 مگابایت)، mindepth- و maxdepth- که حداقل و حداقل عمق دایرکتوری را مشخص می کند، exec- که دستوری را روی فایل های پیدا شده اجرا می کند و ... اشاره کرد.

- دستور head : [۲] می تواند ابتدای یک فایل را پرینت کند. فرمت آن به صورت

## head [options] [file]

است، که از جمله آپشن های آن می توان به c- که تعداد بایت هایی که کاربر می خواهد از ابتدای فایل نمایش داده شود را مشخص می کند، و n- که تعداد خط هایی که کاربر می خواهد از ابتدای فایل نمایش داده شود مشخص می کند.

• دستور tail : [۳] این دستور مانند دستور head است، فقط با این تفاوت که انتهای فایل ها را نشان می دهد. آپشن هایی که در بالا برای head معرفی کردیم، برای این دستور نیز به همان صورت وجود دارند، اما این دستور تعدادی آپشن اضافه تر هم دارد. مهم ترین آنها آپشن f- است که با صورت بی درنگ و به طور دائم فایل را بررسی کرده و هرگاه چیزی به آن اضافه شود، آن را چاپ می کند.

• دستور touch : [۴] این دستور برای ساخت فایل و/یا برای تغییر ویژگی های زمانی فایل ها (مانند access time یا modified time) به کار می رود. اگر آپشنی به کار نرود، این دستور فایلی را با atime و mtime فعلی می سازد. آپشن های a- و m- به ترتیب برای تغییر زمان atime و mtime فایل هایی که وجود دارند نیز می توانند به کار بیایند.

• دستور wc : [۵] این دستور برای شمارش تعداد خطوط/کلمات/کاراکترهای یک فایل می توان به کار برود.  
دستور آن به فرم

## wc [options] [file]

است. اگر آپشنی به کار نرود، به ترتیب تعداد خطوط، تعداد کلمات، و تعداد بایت های آن فایل را چاپ می کند. می توان به ترتیب از آپشن های l- و w- و c- استفاده کرد تا به ترتیب فقط یکی از آن سه عدد چاپ شود.

این دستور دارای آپشن های دیگری نیز است که اطلاعات مشابهی را می دهد. برای مثال m- تعداد کاراکترهای فایل را نمایش می دهد، و L- طول بزرگترین خط را نشان می دهد.

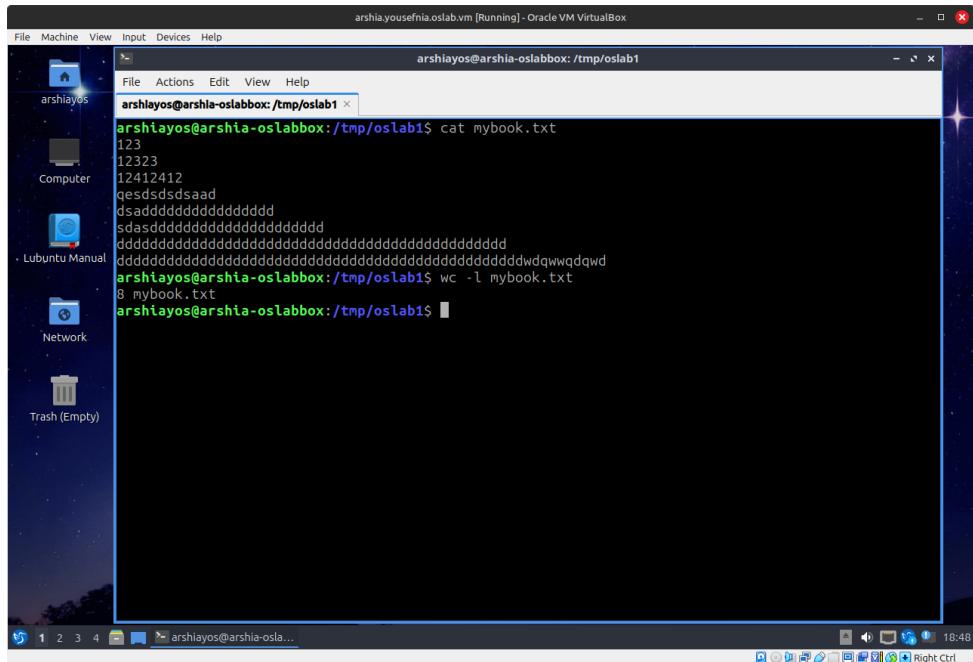
• دستور kill : [۶] این دستور سیگنال مورد نظر کاربر را برای پردازه ها می فرستد و معمولا برای متوقف کردن پردازه ها به کار می رود. معمولا به فرمت زیر است:

## kill [-s signal | --signal signal | -signal] pid

برای مثال 1234 kill سیگنال TERM را برای پردازه 1234 می فرستد. همچنین 1234- kill سیگنال SIGKILL را برای پردازه می فرستد و آن را force kill می کند.

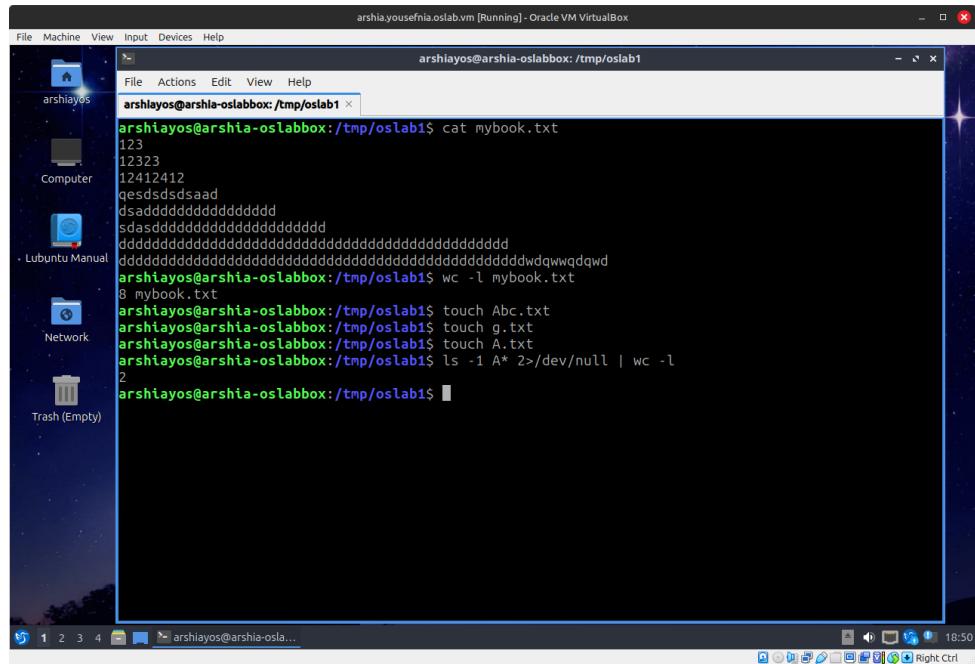
حال با کمک دستوراتی که از این بخش و بخش‌های قبل بررسی کردیم، خواسته‌های صورت آزمایش را انجام می‌دهیم.

۱. برای این کار می‌توان از دستور `wc` با آپشن `-l` استفاده کرد. یک نمونه از اجرای آن را در شکل زیر می‌توانید مشاهده کنید.



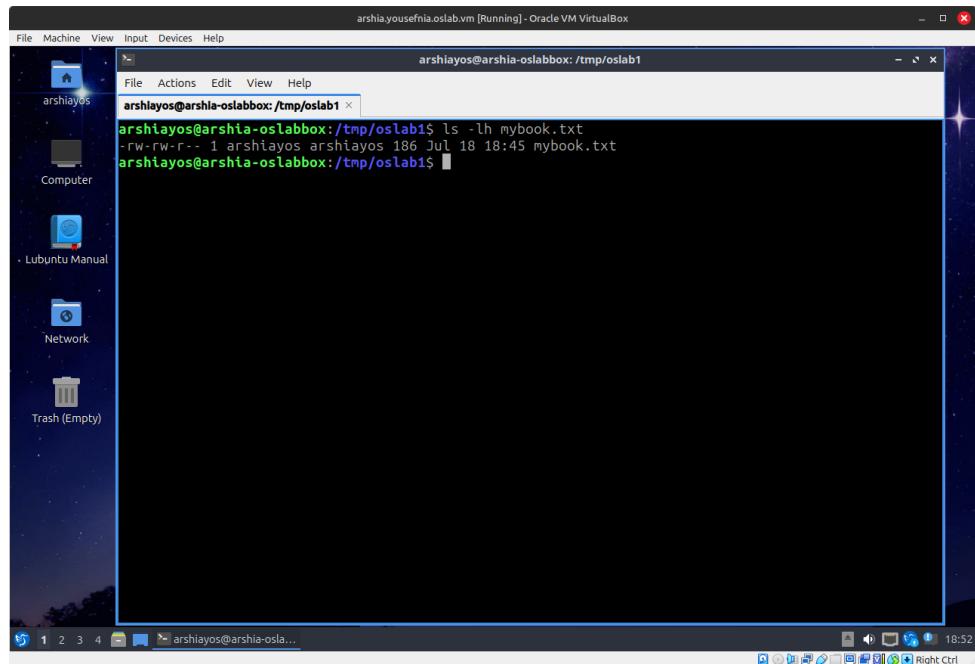
شکل ۲۴: پیدا کردن تعداد خطوط در یک فایل متنه

۲. می‌دانیم که دستور ls نام فایل‌ها را نمایش می‌دهد. اگر از آپشن -1 استفاده کنیم، در هر خط فقط نام یک فایل را نشان می‌دهد. همچنین اگر جلوی ls عبارت A\* باشد، به صورت wildcard فقط فایل‌هایی که نامشان با A شروع می‌شود را نشان می‌دهد. پس اگر خروجی آن را به wc تا تعداد خطوطش را محاسبه کند، به خواسته‌ی خود می‌رسیم. تنها نکته‌ای که باقی می‌ماند این است که ls ممکن است ارور خروجی دهد. پس اگر خطاهای را به ورودی WC ندهیم، هیچگاه مشکلی پیش نمی‌آید. پس دستور \*-ls -A- خواسته‌ی سوال را انجام می‌دهد. در شکل زیر می‌توان خروجی آن را مشاهده کرد.



شکل ۲۵: پیدا کردن تعداد فایل هایی که با حرف A شروع می شوند

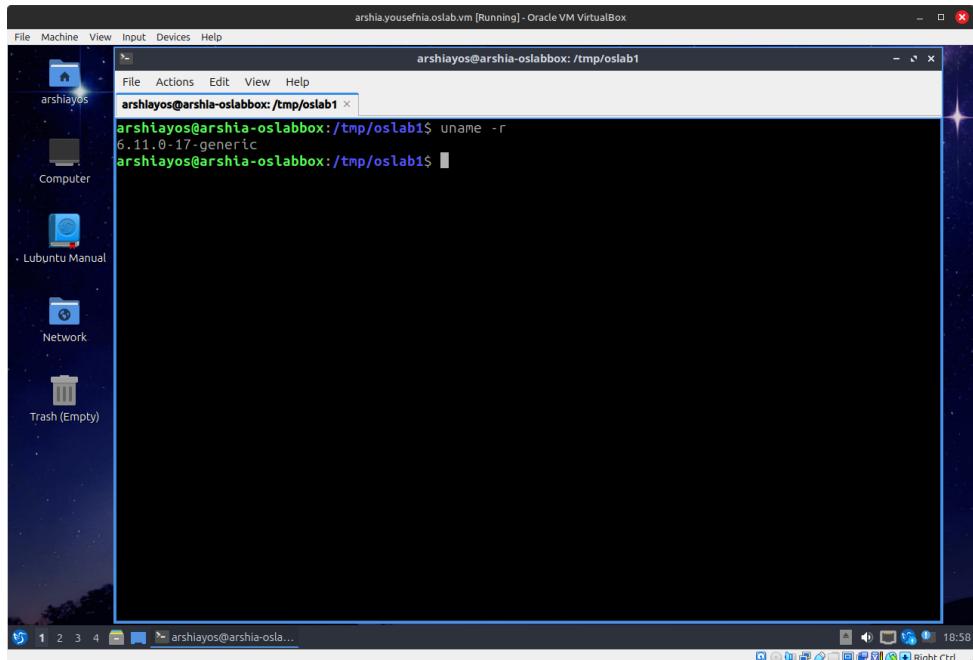
۳. همانطور که در بخش‌های قبل گفتیم، دستور ls -lh اطلاعاتی درمورد فایل‌ها (ازجمله حجمشان) را به ما می‌دهد. پس با دستور mybook.txt -lh می‌توانیم به خواسته‌ی خود برسیم.



## شکل ۲۶: نمایش حجم یک فایل

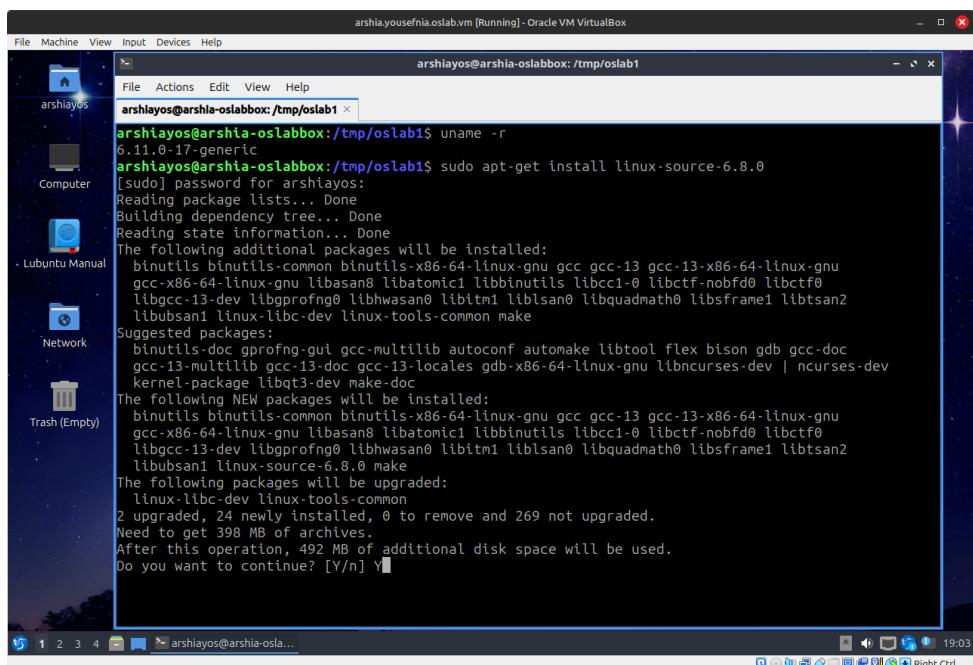
## ۲.۱ اعمال تغیرات و کامپایل مجدد هسته‌ی سیستم عامل

ایندا یا دستور `-r` `uname` نسخه‌ی هسته‌ی فعلی را نشان می‌دهیم.



شکل ۲۷: نمایش نسخه‌ی هسته قبل از کامپایل

۱. مطابق شکل زیر، کد منبع هسته را دریافت می‌کنیم.



شکل ۲۸: دریافت کد هسته

سپس مطابق دو شکل زیر، ابزارهای لازم برای کامپایل هسته را دریافت می‌کنیم.

```

arshiyos@arshia-oslabbbox: /tmp/oslab1
Setting up binutils-x86_64-linux-gnu (2.42-4ubuntu2.5) ...
Setting up gcc-13-x86_64-linux-gnu (13.3.0-6ubuntu2~24.04) ...
Setting up binutils (2.42-4ubuntu2.5) ...
Setting up gcc-13 (13.3.0-6ubuntu2~24.04) ...
Setting up linux-source-6.8.0 (6.8.0-64.67) ...
Setting up gcc-x86_64-linux-gnu (4:13.2.0-7ubuntu1) ...
Setting up gcc (4:13.2.0-7ubuntu1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
arshiyos@arshia-oslabbbox: /tmp/oslab1$ apt-get install build-essential fakeroot
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
arshiyos@arshia-oslabbbox: /tmp/oslab1$ sudo apt-get install build-essential fakeroot
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dpkg-dev g++-13 g++-13-x86_64-linux-gnu g++-x86_64-linux-gnu libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libfakeroot libstdc++-13-dev
  lto-disabled-list
Suggested packages:
  debian-keyring g++-multilib g++-13-multilib gcc-13-doc libstdc++-13-doc
The following NEW packages will be installed:
  build-essential dpkg-dev fakeroot g++ g++-13 g++-13-x86_64-linux-gnu g++-x86_64-linux-gnu
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libfakeroot
  libstdc++-13-dev lto-disabled-list
0 upgraded, 13 newly installed, 0 to remove and 269 not upgraded.
Need to get 15.9 MB of archives.
After this operation, 59.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] 

```

شکل ۲۹: دریافت ابزارهای لازم برای کامپایل هسته (بخش اول)

```

arshiyos@arshia-oslabbbox: /tmp/oslab1
arshiyos@arshia-oslabbbox: /tmp/oslab1$ sudo apt-get install git fakeroot ncurses-dev xz-utils libssl-dev bc libelf-dev bison
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'libncurses-dev' instead of 'ncurses-dev'
fakeroot is already the newest version (1.33-1).
bc is already the newest version (1.07.1-3ubuntu4).
bc set to manually installed.
The following additional packages will be installed:
  git-man libdw164 libelf1 libfl-dev libl2l liblzma5 libssl3t64 libzstd-dev m4 openssl
Suggested packages:
  bison-doc flex-doc git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn ncurses-doc libssl1-doc m4-doc
The following NEW packages will be installed:
  bison flex libelf-dev libfl-dev libncurses-dev libssl-dev libzstd-dev m4
The following packages will be upgraded:
  git git-man libdw164 libelf1 libfl1 liblzma5 libssl3t64 openssl xz-utils
8 upgraded, 9 newly installed, 0 to remove and 261 not upgraded.
Need to get 13.0 MB of archives.
After this operation, 21.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

شکل ۳۰: دریافت ابزارهای لازم برای کامپایل هسته (بخش دوم)

۲. سپس مطابق شکل زیر، کدهای هسته را در یک پوشه بازگشایی می‌کنیم.

```

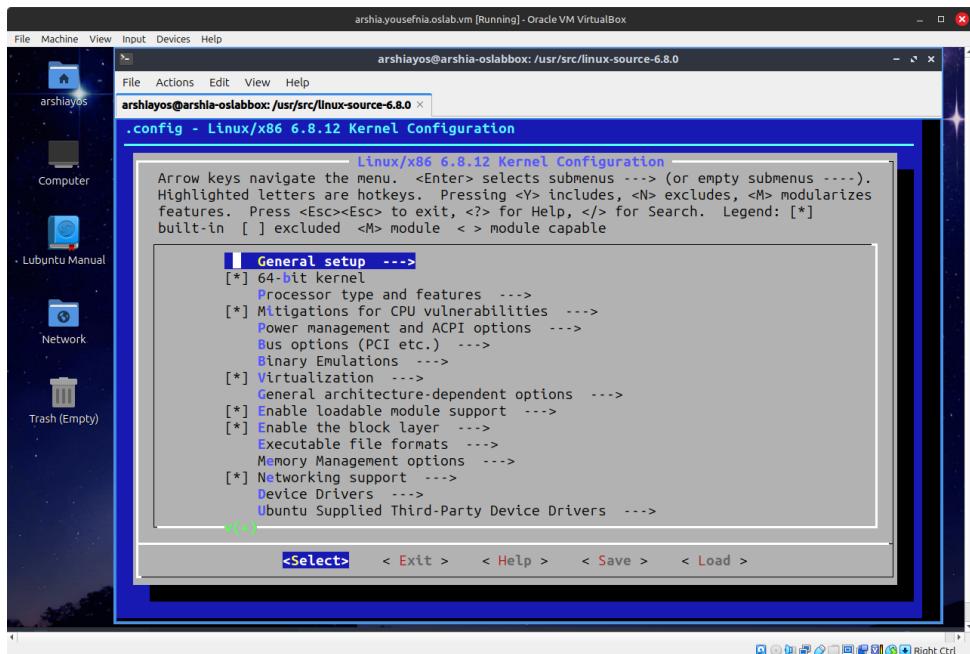
arshayos@arshia-oslabbox: /usr/src/linux-source-6.8.0
arshayos@arshia-oslabbox: /usr/src/linux-source-6.8.0 $ ls
linux-headers-6.11.0-17-generic linux-source-6.8.0
linux-hwe-6.11-headers-6.11.0-17 linux-source-6.8.0.tar.bz2
arshayos@arshia-oslabbox: /usr/src $ sudo tar xjf linux-source-6.8.0.tar.bz2
arshayos@arshia-oslabbox: /usr/src $ ls
linux-headers-6.11.0-17-generic linux-source-6.8.0
linux-hwe-6.11-headers-6.11.0-17 linux-source-6.8.0.tar.bz2
arshayos@arshia-oslabbox: /usr/src $ cd linux-source-6.8.0
arshayos@arshia-oslabbox: /usr/src/linux-source-6.8.0 $ cp /boot/config-$uname -r .config
cp: cannot create regular file '.config': Permission denied
arshayos@arshia-oslabbox: /usr/src/linux-source-6.8.0 $ sudo cp /boot/config-$uname -r .config
arshayos@arshia-oslabbox: /usr/src/linux-source-6.8.0 $ ls
arch drivers io_uring linux-source-6.8.0.tar.bz2 samples usr
block dropped.txt ipc MAINTAINERS scripts virt
certs fs Kconfig mm security
COPYING generic.depmod.log kernel net sound
CREDITS generic.inclusion-list.log lib README tools
crypto include ltb LICENSES rust ubuntu
Documentation init LICENSES samples Ubuntu.md
arshayos@arshia-oslabbox: /usr/src/linux-source-6.8.0 $ ls -a
.
..
crypto init io_uring MAINTAINERS scripts security
Documentation ipc Makefile sound
arch drivers kernel net ubuntu
block dropped.txt Kbuild mm tools
certs fs Kconfig net Ubuntu.md
.CONGRATULATIONS generic.depmod.log lib rust usr
CREDITS include LICENSES samples virt
arshayos@arshia-oslabbox: /usr/src/linux-source-6.8.0 $

```

شکل ۳۱: بازگشایی کدهای هسته و ورود به آن

## ۲،۲ فعالیت‌ها (ادامه‌ی کامپایل)

در این بخش توضیح کوتاهی درمورد ادامه‌ی مراحل کامپایل داده و شکل‌های مربوط به آن بخشن را می‌گذاریم.  
در ادامه‌ی مراحل کامپایل، ابتدا باید configuration سیستم عامل را برای کامپایل تنظیم کنیم. یکی از روش‌های پر استفاده، دستور make menuconfig است که دارای محیط کاربری برای تنظیم config است.



شکل ۳۲: محیط make menuconfig

اما از آنجایی که ما صرفا داریم ورژن هسته‌ی لینوکس را تغییر می‌دهیم، یک روش ساده‌تر، استفاده از

است که از config قبلی استفاده می‌کند و فقط درباره موارد جدید config از کاربر سوال می‌پرسد. ما هم از config قبلی یک backup گرفته و با این دستور، فایل config را آماده می‌کنیم.

```
arshiyos@arshia-oslabbox:~/src/linux-source-6.8.0$ cp /boot/config-$(uname -r) .config
arshiyos@arshia-oslabbox:~/src/linux-source-6.8.0$ make oldconfig
HOSTCC scripts/basic/fixedep
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
.config:273:warning: symbol value 'n' invalid for BASE_SMALL
.config:4776:warning: symbol value 'm' invalid for SERIAL_SC16IS7XX_I2C
.config:4777:warning: symbol value 'm' invalid for SERIAL_SC16IS7XX_SPI
*
* Restart config...
*
* Configure standard kernel features (expert users)
*
Configure standard kernel features (expert users) (EXPERT) [Y/n/?] y
Enable 16-bit UID system calls (UID16) [Y/n/?] y
Multiple users, groups and capabilities support (MULTIUSER) [Y/n/?] y
setmask/ssetmask syscalls support (SGETMASK SYSCALL) [Y/n/?] y
Sysfs syscall support (SYSFS_SYSCALL) [Y/n/?] y
open by fhandle syscalls (FHANDLE) [Y/n/?] y
```

شکل ۳۳: استفاده از make oldconfig

سپس با دستور make -j\$(nproc) هسته را کامپایل می‌کنیم.

```
arshiyos@arshia-oslabbox:~/src/linux-source-6.8.0$ make -j$(nproc)
Test module for correctness and stress of objpool (TEST_OBJPOOL) [N/m/?] n
#
# configuration written to .config
#
arshiyos@arshia-oslabbox:~/src/linux-source-6.8.0$ make -j$(nproc)
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
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
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_32.h
HOSTCC arch/x86/tools/relocs_32.o
SYSHDR arch/x86/include/generated/uapi/asm/unistd_64.h
SYSHDR arch/x86/include/generated/uapi/asm/unistd_x32.h
SYSTBL arch/x86/include/generated/asm/syscalls_32.h
SYSHDR arch/x86/include/generated/asm/unistd_32_i32.h
SYSHDR arch/x86/include/generated/asm/unistd_64_x32.h
SYSTBL arch/x86/include/generated/asm/syscalls_64.h
HYPERCALLS arch/x86/include/generated/asm/xen-hypercalls.h
HOSTCC arch/x86/tools/relocs_64.o
HOSTCC arch/x86/tools/relocs_common.o
```

شکل ۳۴: کامپایل مجدد هسته

پس از کامپایل، باید هسته جدید را نصب کنیم. برای این کار ابتدا ماثولوها را با دستور make modules\_install نصب کرده و سپس با دستور make install هسته را نصب کرد.

```

arshayos@arshia-oslabbbox: /usr/src/linux-source-6.8.0$ sudo make modules_install
[sudo] password for arshayos:
SYMLINK /lib/modules/6.8.12/build
INSTALL /lib/modules/6.8.12/modules.order
INSTALL /lib/modules/6.8.12/modules.builtin
INSTALL /lib/modules/6.8.12/modules.builtin.modinfo
INSTALL /lib/modules/6.8.12/kernel/arch/x86/events/amd/amd-uncore.ko
SIGN /lib/modules/6.8.12/kernel/arch/x86/events/amd/amd-uncore.ko
INSTALL /lib/modules/6.8.12/kernel/arch/x86/events/intel/intel-cstate.ko
SIGN /lib/modules/6.8.12/kernel/arch/x86/events/intel/intel-cstate.ko
INSTALL /lib/modules/6.8.12/kernel/arch/x86/events/rapl.ko
SIGN /lib/modules/6.8.12/kernel/arch/x86/events/rapl.ko
INSTALL /lib/modules/6.8.12/kernel/arch/x86/kernel/cpu/mce/inject.ko
SIGN /lib/modules/6.8.12/kernel/arch/x86/kernel/cpu/mce/inject.ko
INSTALL /lib/modules/6.8.12/kernel/arch/x86/kernel/msr.ko
SIGN /lib/modules/6.8.12/kernel/arch/x86/kernel/msr.ko
INSTALL /lib/modules/6.8.12/kernel/arch/x86/kernel/cpuid.ko
SIGN /lib/modules/6.8.12/kernel/arch/x86/kernel/cpuid.ko
INSTALL /lib/modules/6.8.12/kernel/arch/x86/crypto/twofish-x86_64.ko
SIGN /lib/modules/6.8.12/kernel/arch/x86/crypto/twofish-x86_64.ko
INSTALL /lib/modules/6.8.12/kernel/arch/x86/crypto/twofish-x86_64-3way.ko
SIGN /lib/modules/6.8.12/kernel/arch/x86/crypto/twofish-x86_64-3way.ko
INSTALL /lib/modules/6.8.12/kernel/arch/x86/crypto/twofish-avx-x86_64.ko
SIGN /lib/modules/6.8.12/kernel/arch/x86/crypto/twofish-avx-x86_64.ko

```

شکل ۳۵: نصب مازول‌های هسته

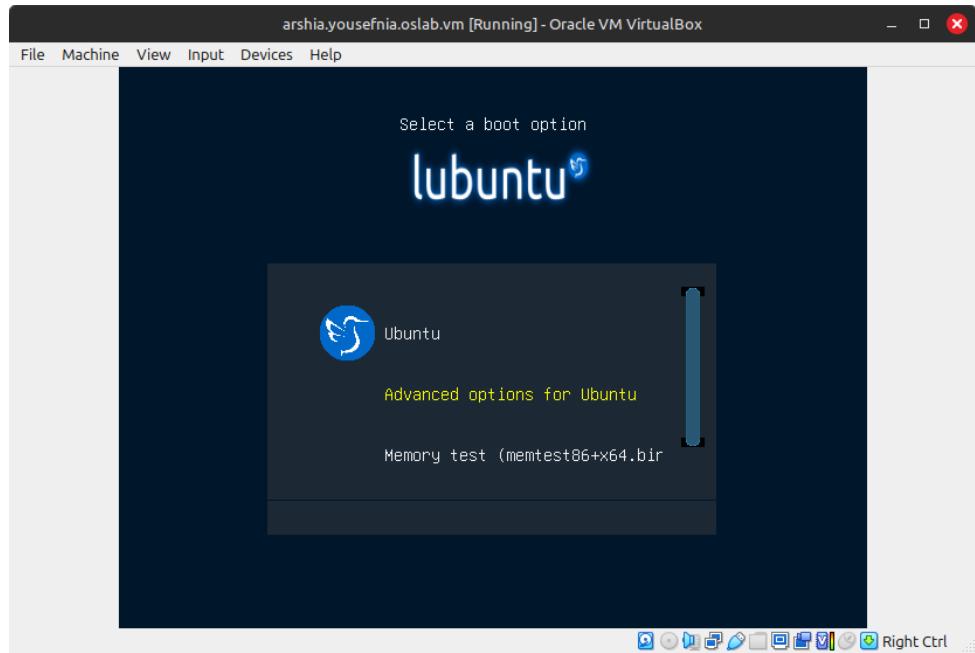
```

arshayos@arshia-oslabbbox: /usr/src/linux-source-6.8.0$ sudo make install
INSTALL /lib/modules/6.8.12/kernel/net/rtr/rtr-mhi.ko
SIGN /lib/modules/6.8.12/kernel/net/rtr/rtr-mhi.ko
INSTALL /lib/modules/6.8.12/kernel/virt/lib/irqbypass.ko
SIGN /lib/modules/6.8.12/kernel/virt/lib/irqbypass.ko
INSTALL /lib/modules/6.8.12/kernel/ubuntu/ubuntu-host/ubuntu-host.ko
SIGN /lib/modules/6.8.12/kernel/ubuntu/ubuntu-host/ubuntu-host.ko
DEPMOD /lib/modules/6.8.12
INSTALLED /boot
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 6.8.12 /boot/vmlinuz-6.8.12
update-initramfs: Generating /boot/initrd.img-6.8.12
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 6.8.12 /boot/vmlinuz-6.8.12
run-parts: executing /etc/kernel/postinst.d/dx-update-initrd-links 6.8.12 /boot/vmlinuz-6.8.12
I: /boot/initrd.img is now a symlink to initrd.img-6.8.12
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 6.8.12 /boot/vmlinuz-6.8.12
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/lubuntu-grub-theme.cfg'
Generating grub configuration file ...
Found theme: /usr/share/grub/themes/lubuntu-grub-theme/theme.txt
Found linux image: /boot/vmlinuz-6.11.0-17-generic
Found initrd image: /boot/initrd.img-6.11.0-17-generic
Found linux image: /boot/vmlinuz-6.8.12
Found initrd image: /boot/initrd.img-6.8.12
Found memtest86+x64 image: /boot/memtest86+x64.bin
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
Adding boot menu entry for UEFI Firmware Settings ...
done
arshayos@arshia-oslabbbox: /usr/src/linux-source-6.8.0$ 

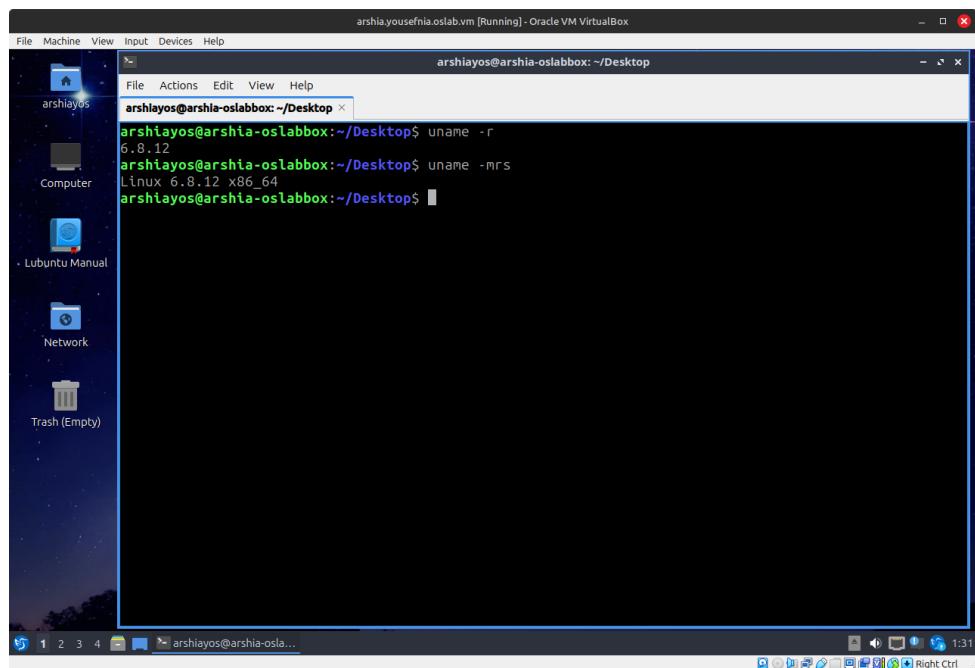
```

شکل ۳۶: نصب image هسته

درنهایت باید سیستم را reboot کنیم. هنگام boot کردن، باید نسخه‌ای که به تازگی کامپایل کردہ‌ایم را انتخاب کرده و اجرا کنیم.



شکل ۳۷: انتخاب نسخه جدید هنگام boot



شکل ۳۸: بررسی درستی کامپایل و نصب با دستور `uname -r`

## مراجع

- [١] GNU Project. cut invocation. Accessed: .2025-07-28 .2024 URL: [https://www.gnu.org/software/coreutils/manual/html\\_node/cut-invocation.html](https://www.gnu.org/software/coreutils/manual/html_node/cut-invocation.html).
- [٢] GNU Project. head invocation. Accessed: .2025-07-28 .2024 URL: [https://www.gnu.org/software/coreutils/manual/html\\_node/head-invocation.html](https://www.gnu.org/software/coreutils/manual/html_node/head-invocation.html).
- [٣] GNU Project. tail invocation. Accessed: .2025-07-28 .2024 URL: [https://www.gnu.org/software/coreutils/manual/html\\_node/tail-invocation.html](https://www.gnu.org/software/coreutils/manual/html_node/tail-invocation.html).
- [٤] GNU Project. touch invocation. Accessed: .2025-07-28 .2024 URL: [https://www.gnu.org/software/coreutils/manual/html\\_node/touch-invocation.html](https://www.gnu.org/software/coreutils/manual/html_node/touch-invocation.html).
- [٥] GNU Project. wc invocation. Accessed: .2025-07-28 .2024 URL: [https://www.gnu.org/software/coreutils/manual/html\\_node/wc-invocation.html](https://www.gnu.org/software/coreutils/manual/html_node/wc-invocation.html).
- [٦] GNU Project. kill invocation. Accessed: .2025-07-28 .2024 URL: [https://www.gnu.org/software/coreutils/manual/html\\_node/kill-invocation.html](https://www.gnu.org/software/coreutils/manual/html_node/kill-invocation.html).