



Sharif-OS-Lab /

summer1403-1-2-99101087_99100422



<> Code

Issues 2

Pull requests

Actions

Projects

Security

Insights

New issue

[Jump to bottom](#)

Session 2 Report #2

Open

34 tasks done

BozorgmehrZia opened this issue last week · 0 comments

Assignees



Labels

documentation

BozorgmehrZia commented last week • edited ▼

Team Name: 99101087-99100422

Student Name of member 1: امیررضا آنری

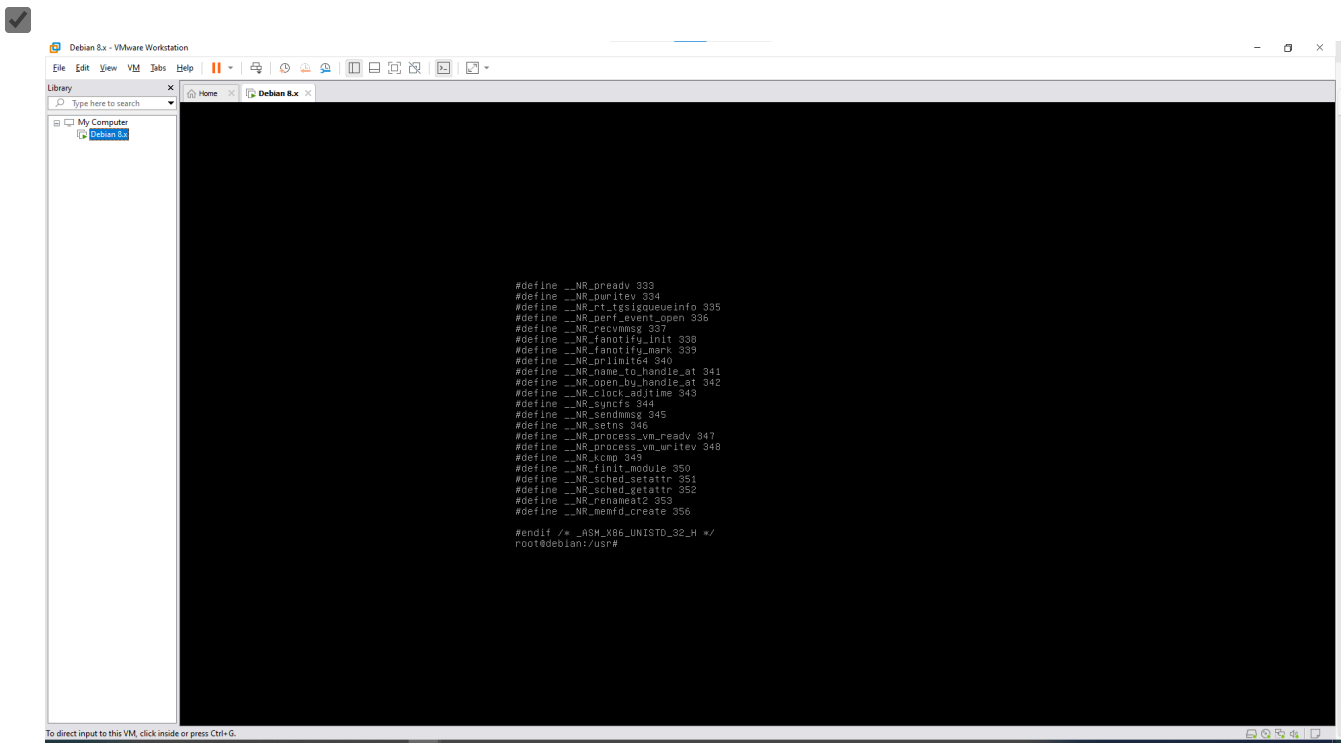
Student No. of member 1: 99101087

Student Name of member 2: بزرگمهر ضیاء

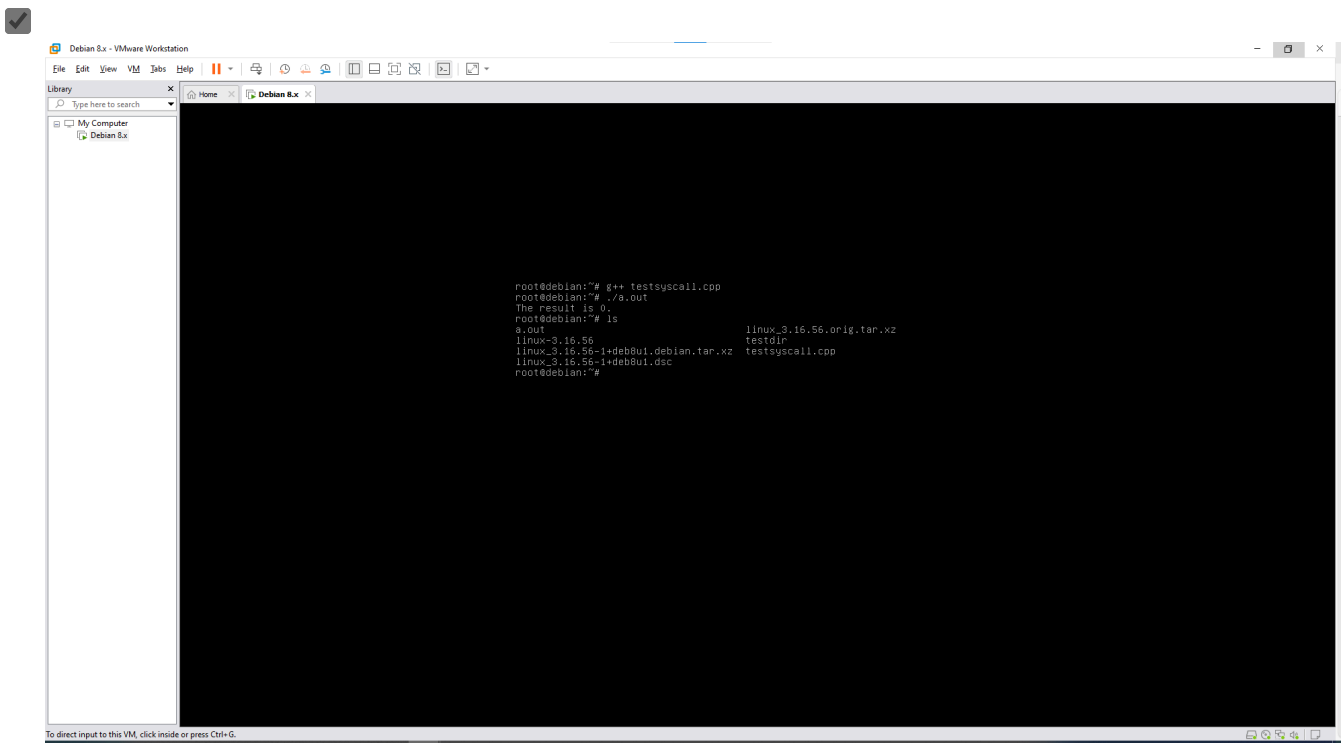
Student No. of member 2: 99100422

☒ Read Session Contents.

Section 2.3.1



Section 2.3.2



- این دستور، یک دایرکتوری با نام داده شده و با پرمیشن 0777 می سازد و در صورت ساخت موفق، 0 را برمی گرداند. 0777 به این معناست که قابل خواندن، قابل نوشتن و قابل اجرا برای کاربر جاری و کاربران داخل و خارج گروه کاربر جاری است.
- این ثابت برابر با شماره تابع سیستمی ساخت پوشه در کرنل لینوکس است که شماره مربوط به آن در سیستم عامل تعریف شده است.

- ✓ ورودی های این تابع، شماره تابع سیستمی و ورودی های مربوط به آن تابع (به طور مثال در اینجا اسم پوشه و پرمیشن آن) هستند. خروجی آن نیز نشان دهنده وضعیت انجام تابع از سمت سیستم عامل است که 0 به معنای موفقیت است. میتوان با اجرای دستور زیر از عملکرد تابع سیستمی اطلاع پیدا کرد:
- ```
man 2 syscall
```

## Section 2.3.3



```
GNU nano 2.2.6 File: testsyscall2.cpp

#include <stdio.h>
#include <sys/stat.h>

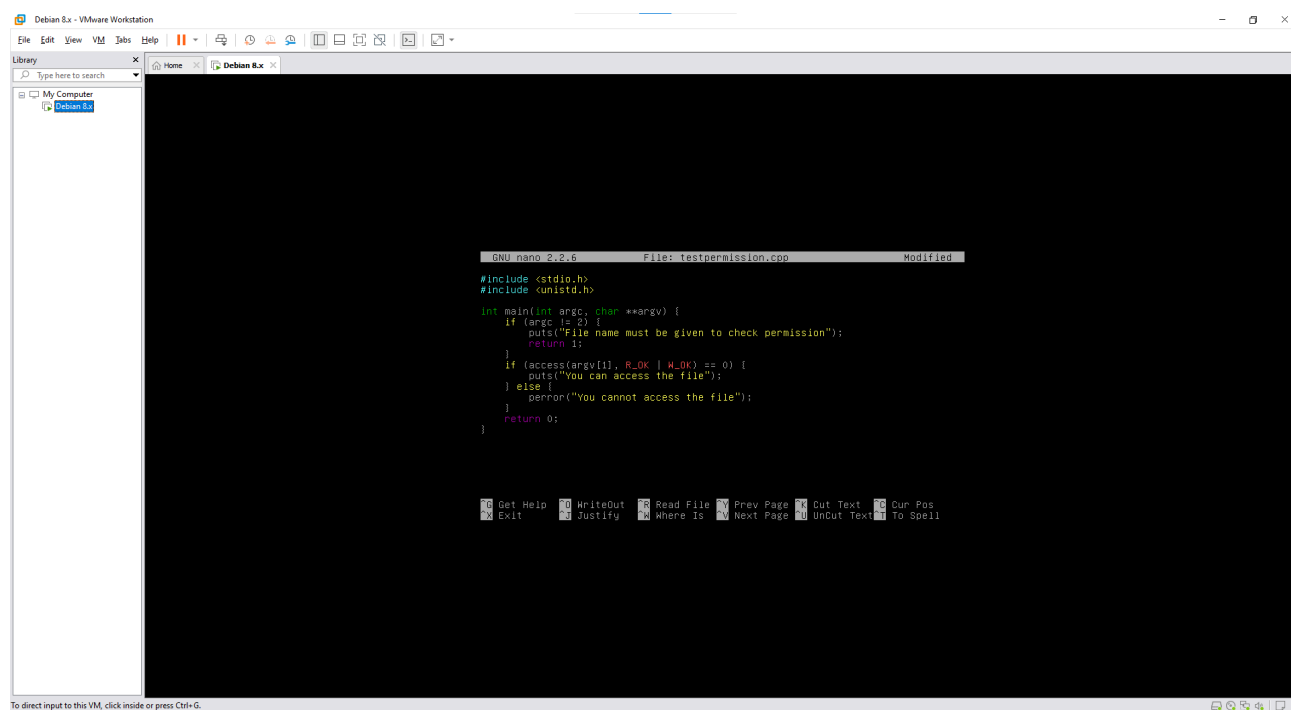
int main() {
 int result = mkdir("testdir", 0777);
 printf("result: %d\n", result);
 return 0;
}
```



```
root@debian:~# g++ testsyscall2.cpp
root@debian:~# ls
a.out linux_3.16.56.orig.tar.xz
linux-3.16.56 testdir
linux_3.16.56-1-deb8u1.debian.tar.xz testsyscall2.cpp
linux_3.16.56-1-deb8u1.dsc testsyscall.cpp
root@debian:~# ./a.out
result: -1
root@debian:~#
```

در اینجا، چون پوشه مورد نظر از قبل وجود داشت، مقدار -1 برگردانده شد.

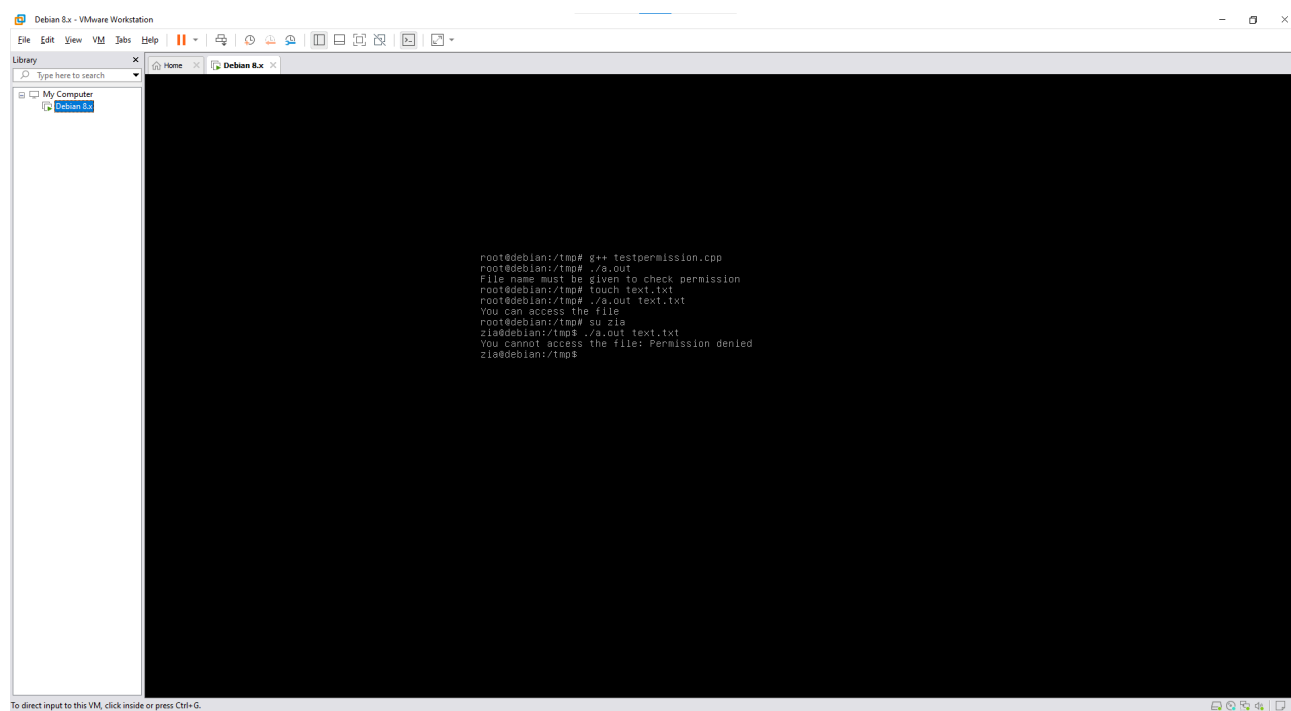
## Section 2.3.4



```
GNU nano 2.2.6 File: testpermission.cpp Modified
#include <stdio.h>
#include <unistd.h>

int main(int argc, char **argv) {
 if (argc != 2) {
 puts("File name must be given to check permission");
 return 1;
 }
 if (access(argv[1], R_OK | W_OK) != 0) {
 puts("You can access the file");
 } else {
 perror("You cannot access the file");
 }
 return 0;
}
```

طبق کد، ابتدا چک می شود که نام فایل به عنوان آرگومان به تابع main داده شده باشد. سپس با صدا زدن تابع سیستمی access چک می شود آیا دسترسی به فایل مورد نظر امکان پذیر است یا خیر.



```
root@debian:/tmp# g++ testpermission.cpp
root@debian:/tmp# ./a.out
File name must be given to check permission
root@debian:/tmp# touch text.txt
root@debian:/tmp# ./a.out text.txt
You can access the file
root@debian:/tmp# su zia
zia@debian:/tmp$./a.out text.txt
You cannot access the file: Permission denied
zia@debian:/tmp$
```

در اینجا، ابتدا کد بدون آرگومان نام تابع اجرا می شود که با خطا مواجه می شود. سپس اسم یک فایل به عنوان آرگومان داده شده و پیغام مورد نظر دریافت می شود. سپس کاربر سیستم عامل به کاربری غیر از root تغییر کرده و کد اجرا شده که با خطای عدم دسترسی مواجه می شود.

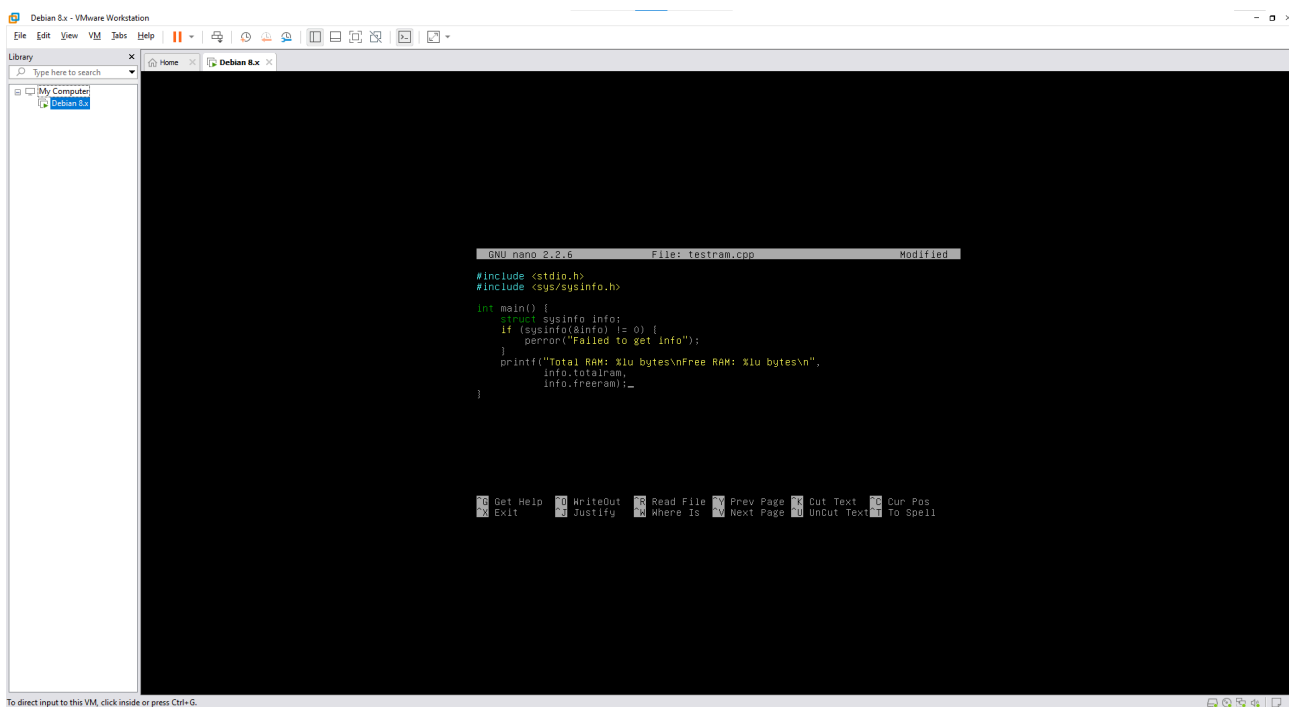


```
GNU nano 2.2.6 File: testur1tefile.cpp Modified
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
int main() {
 int fd = open("oslab2.txt", O_WRONLY | O_CREAT | O_TRUNC, 0666);
 if (fd < 0) {
 perror("Failed to open file oslab2.txt");
 return 1;
 }
 const char *name = "Amir Reza Azari & Bozorgmehr Zia";
 if (write(fd, name, strlen(name)) < 0) {
 perror("Failed to write to file oslab2.txt");
 return 1;
 }
 if (close(fd) < 0) {
 perror("Failed to close file oslab2.txt");
 return 1;
 }
 return 0;
}
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page Uncut Text To Spell
```

طبق کد، ابتدا با تابع سیستمی open، فایل مورد نظر باز می شود و در صورت عدم موفقیت پیغام خطا داده می شود. سپس با تابع سیستمی write، محتوای مورد نظر در فایل نوشته می شود و در صورت عدم موفقیت پیغام خطا داده می شود. سپس با تابع سیستمی close، فایل مورد نظر بسته می شود و در صورت عدم موفقیت پیغام خطا داده می شود.



```
root@debian:/tmp# g++ testur1tefile.cpp
root@debian:/tmp# ./a.out
root@debian:/tmp# cat oslab2.txt
Amir Reza Azari & Bozorgmehr Zia
root@debian:/tmp#
```

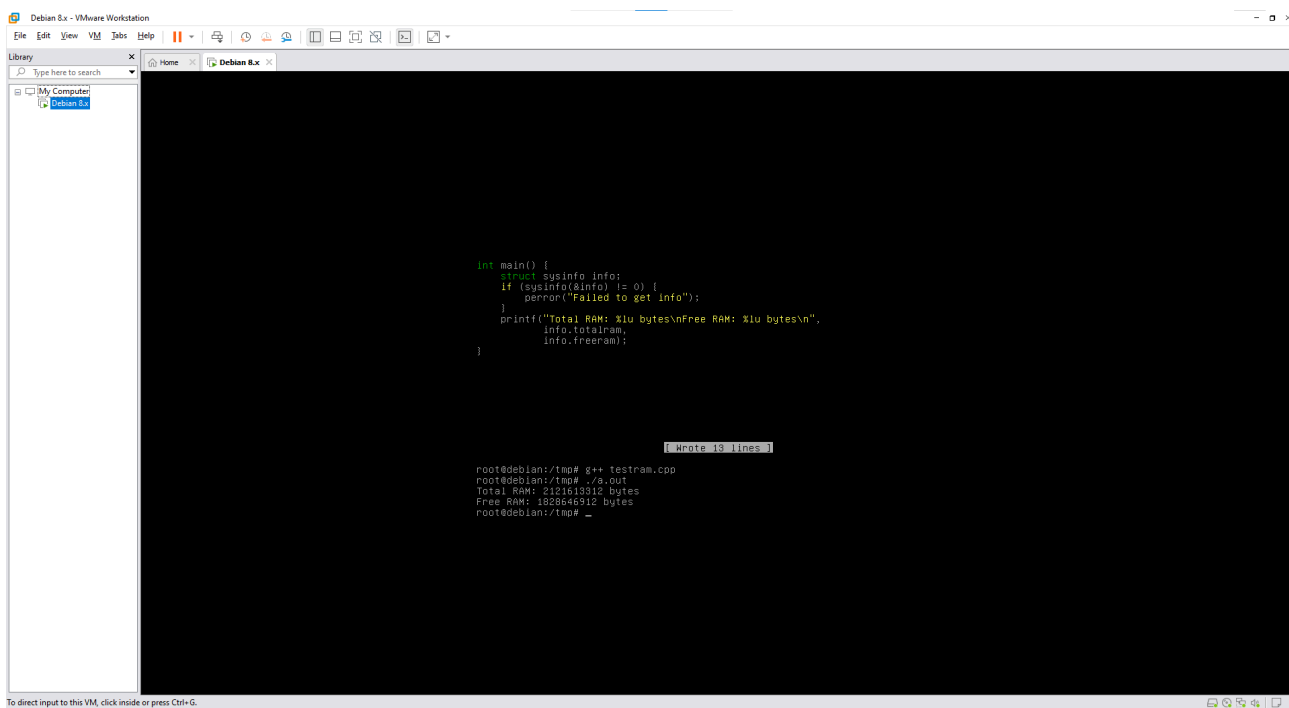


```
GNU nano 2.2.6 File: testram.cpp Modified

#include <stdio.h>
#include <sys/sysinfo.h>

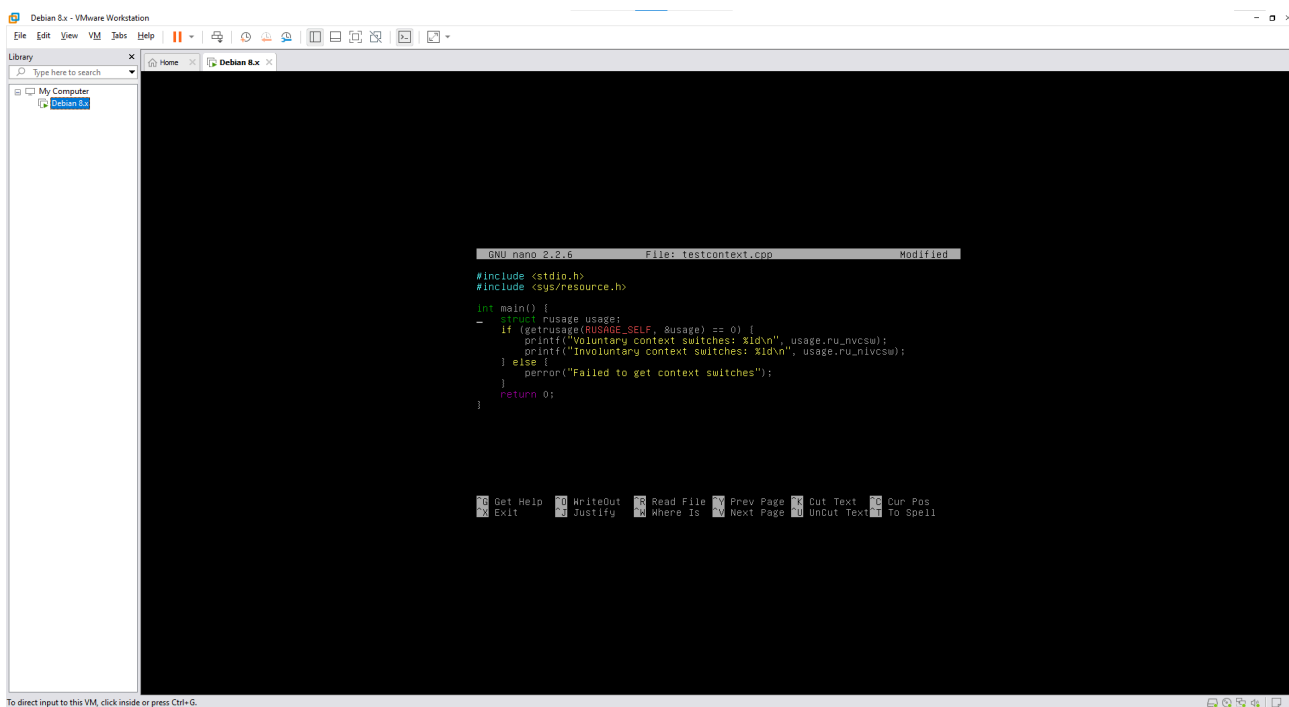
int main() {
 struct sysinfo info;
 if (sysinfo(&info) != 0) {
 perror("Failed to get info");
 }
 printf("Total RAM: %lu bytes\nFree RAM: %lu bytes\n",
 info.totalram,
 info.freeram);
}
```

در این کد از تابع سیستمی `sysinfo` و شیء سیستمی `sysinfo` و ویژگی های `totalram` برای مقدار کل حافظه و `freeram` برای مقدار حافظه خالی استفاده شده است.



```
int main() {
 struct sysinfo info;
 if (sysinfo(&info) != 0) {
 perror("Failed to get info");
 }
 printf("Total RAM: %lu bytes\nFree RAM: %lu bytes\n",
 info.totalram,
 info.freeram);
}
```

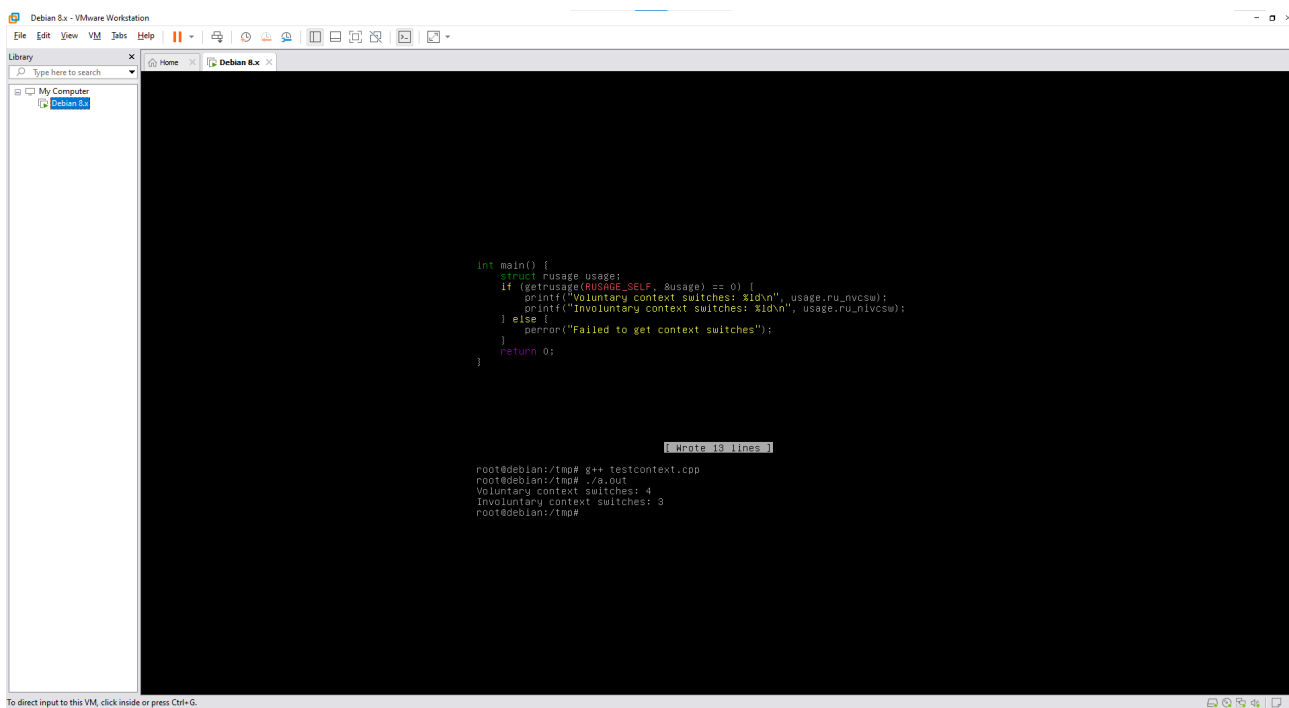
```
root@debian:/tmp# g++ testram.cpp
root@debian:/tmp# ./a.out
Total RAM: 2121613312 bytes
Free RAM: 1828646912 bytes
root@debian:/tmp#
```



```
GNU nano 2.2.6 File: testcontext.cpp Modified
#include <stdio.h>
#include <sys/resource.h>

int main() {
 struct rusage usage;
 if (getrusage(RUSAGE_SELF, &usage) == 0) {
 printf("Voluntary context switches: %ld\n", usage.ru_nvcsw);
 printf("Involuntary context switches: %ld\n", usage.ru_nivcsw);
 } else {
 perror("Failed to get context switches");
 }
 return 0;
}
```

در این کد از تابع سیستمی `getrusage` و شیء سیستمی `rusage` و ویژگی های `ru_nvcsw` برای تعداد `context swith` های داوطلبانه و `ru_nivcsw` برای تعداد `context swith` های غیر داوطلبانه استفاده شده است.



```
int main() {
 struct rusage usage;
 if (getrusage(RUSAGE_SELF, &usage) == 0) {
 printf("Voluntary context switches: %ld\n", usage.ru_nvcsw);
 printf("Involuntary context switches: %ld\n", usage.ru_nivcsw);
 } else {
 perror("Failed to get context switches");
 }
 return 0;
}

root@debian:/tmp# g++ testcontext.cpp
root@debian:/tmp# ./a.out
Voluntary context switches: 4
Involuntary context switches: 3
root@debian:/tmp#
```

## Section 2.3.5

- ✓ Do 13 subtasks from 1 to 13 and add screenshots of each:

Because we logged in with `root` user, we didn't provide an image of it. Also, we skipped the image of change directory to linux source.

We put images for making and installing kernel after changing kernel codes, because we had compiled the kernel for session 1 and we compiled it again to apply changes.



```
root@debian:~/linux-3.16.56# mkdir hello
root@debian:~/linux-3.16.56# ls
arch fs lib README usr
block hello MAINTAINERS REPORTING-BUGS virt
CDPVING include Makefile samples vmlinux
CREDITS init mm scripts vmlinux.o
crypto ipc modules.builtin security
debian kbuild modules.order sound
documentation kconfig Module.symvers system.map
drivers kernel net tools
root@debian:~/linux-3.16.56# cd hello
root@debian:~/linux-3.16.56/hello#
```

Created folder `hello` in kernel code root directory.



```
GNU nano 2.2.6 File: hello.c Modified
#include <linux/kernel.h>

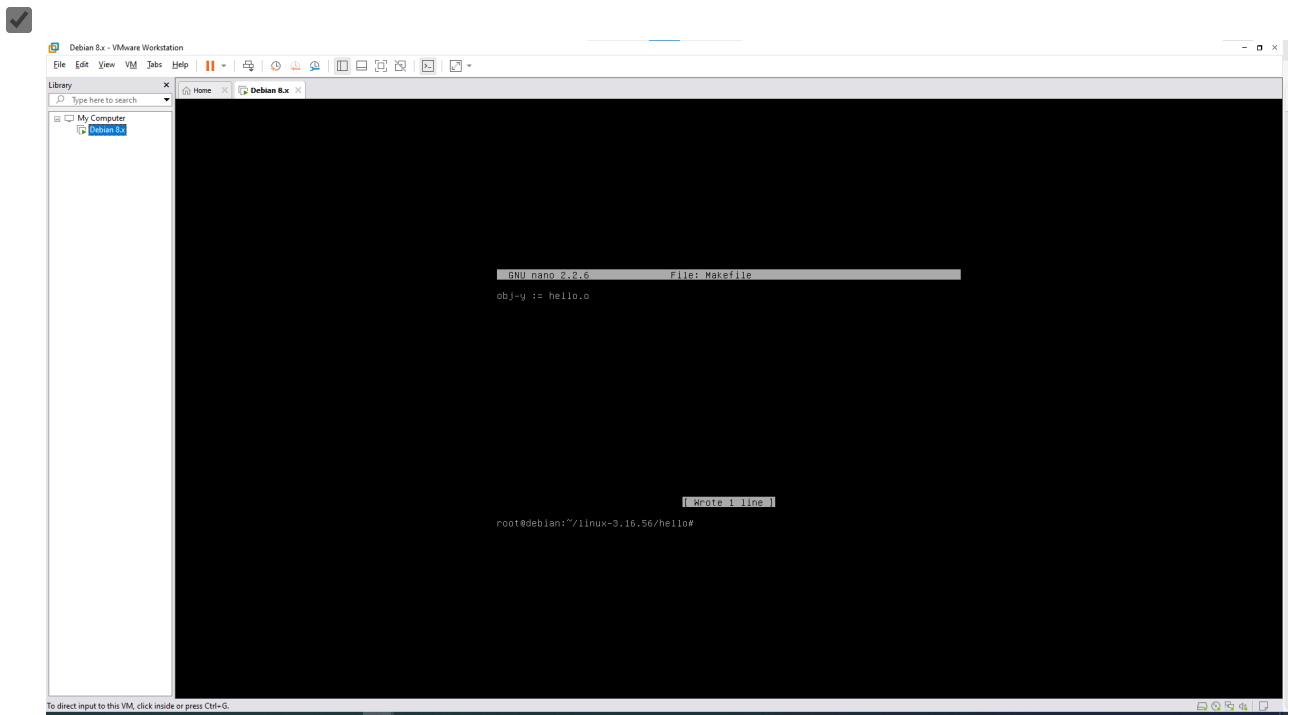
asmlinkage long sys_hello(void) {
 printk("Hello World\n");
 return 0;
}

asmlinkage long sys_add(long a, long b) {
 return a + b;
}

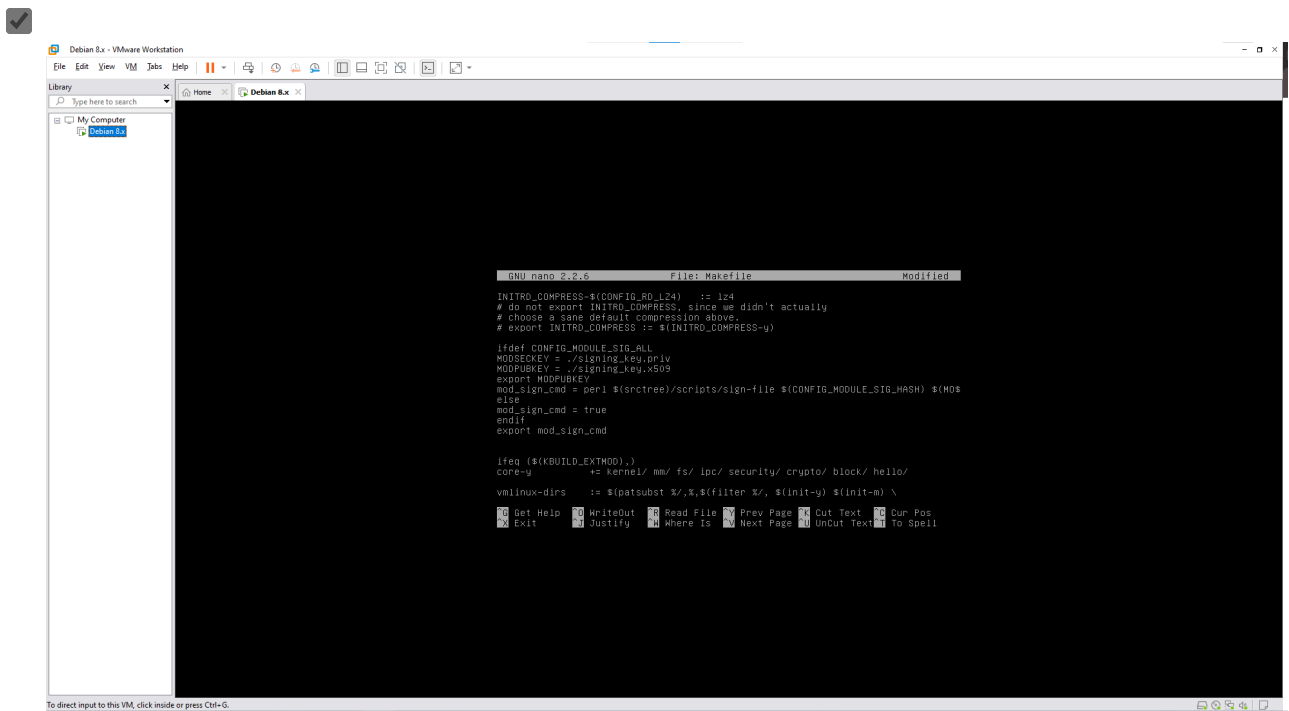
/*
```

Both system calls `hello world` and `adder` were added to the file `hello.c`

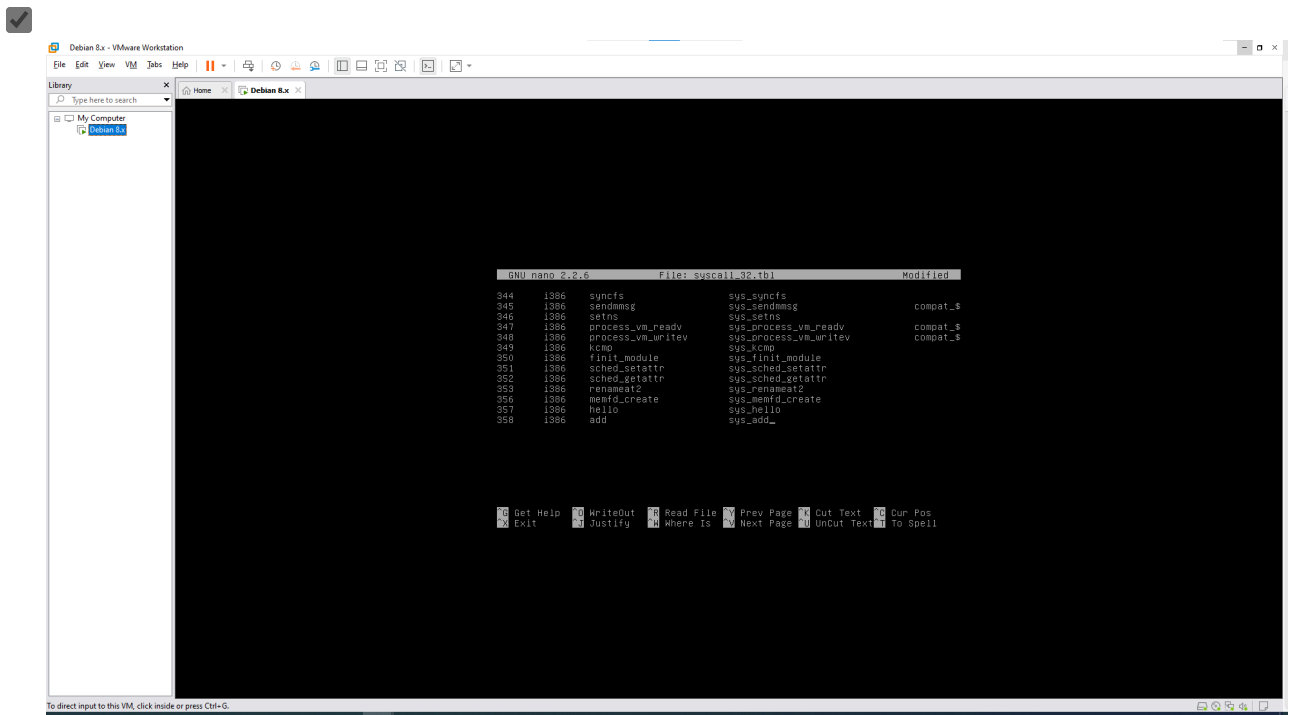




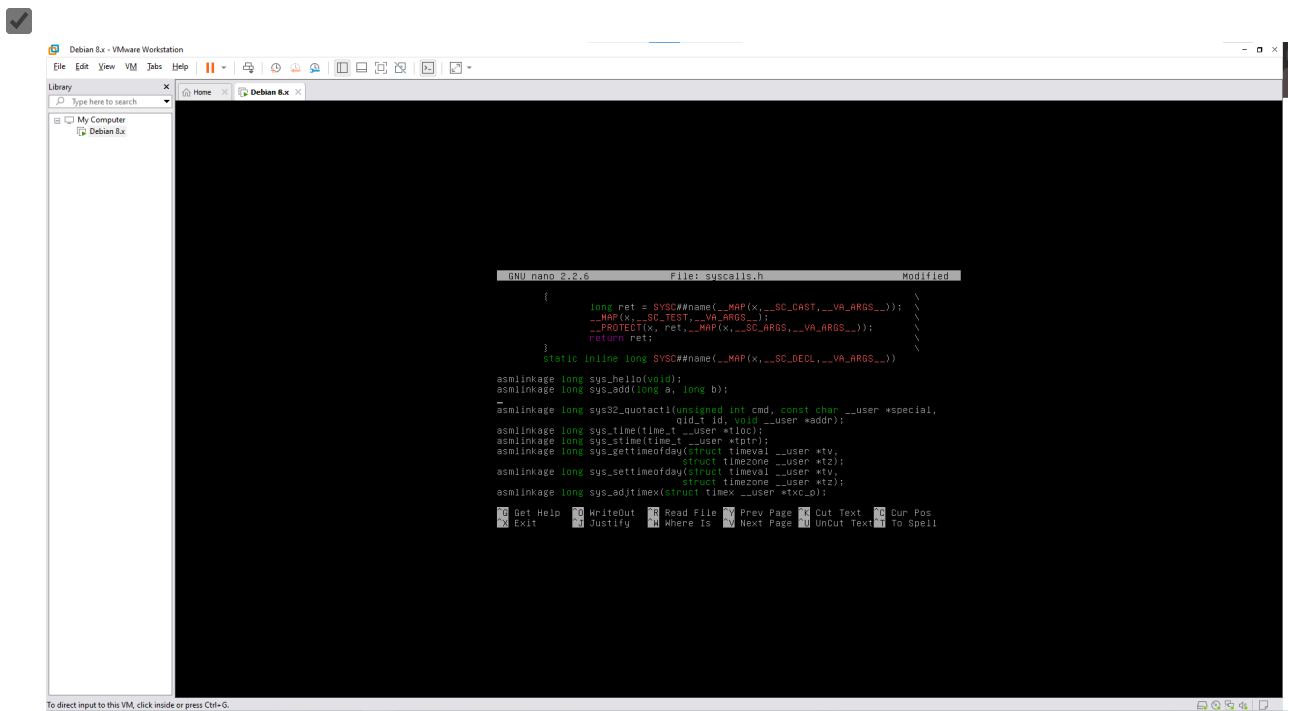
File Makefile was created in the same directory with given content.



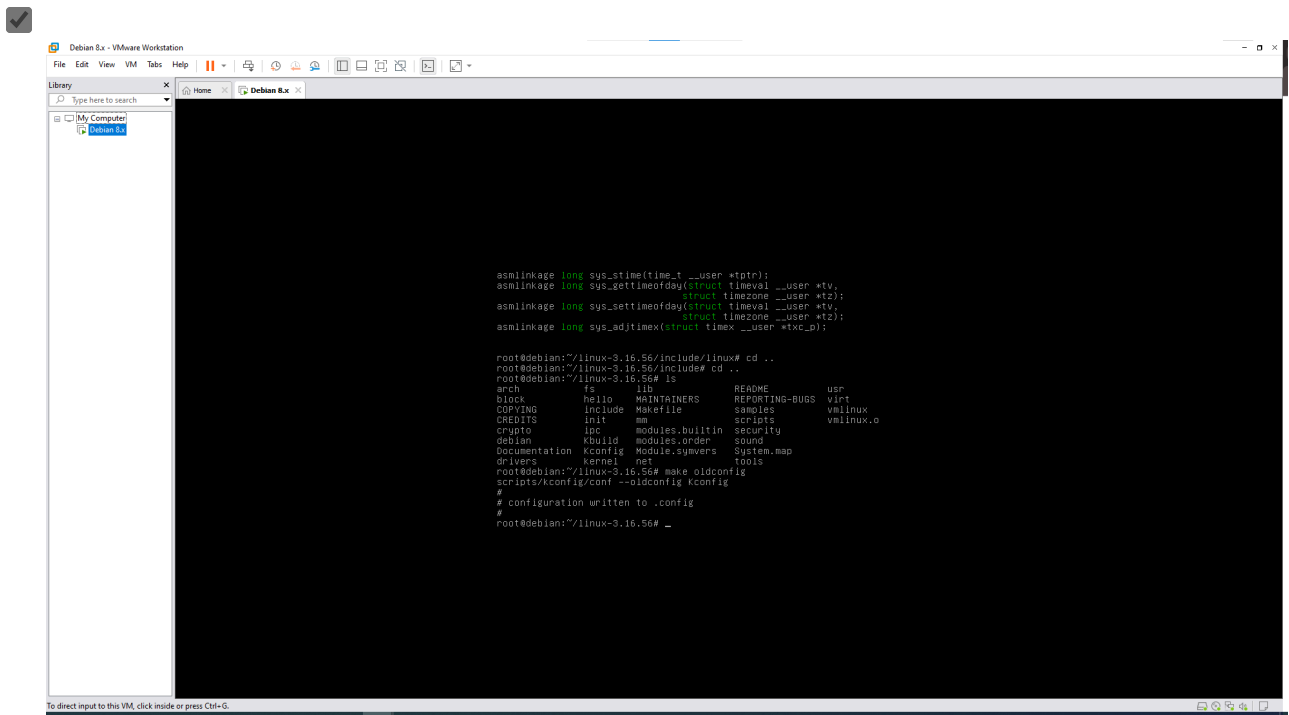
Added hello/ to Makefile in the root.



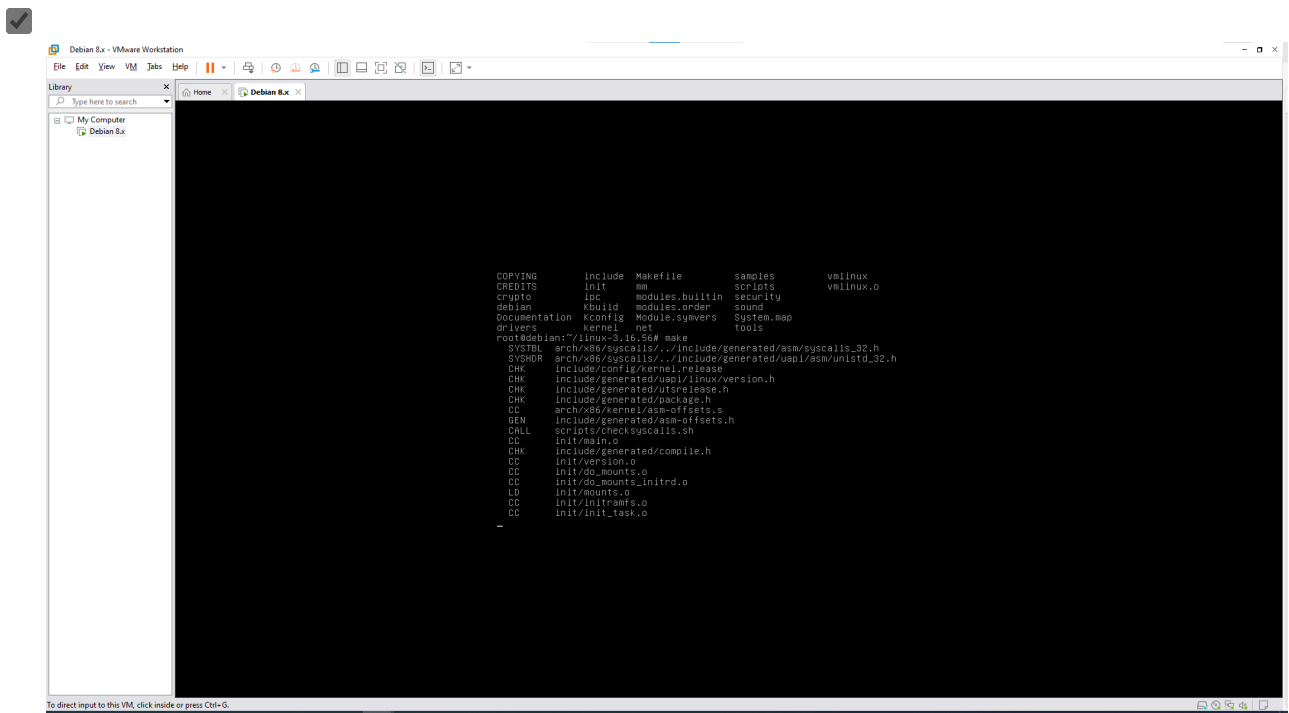
Syscalls for hello and add were added to the file `syscall_32.tbl` with numbers 357 and 358.



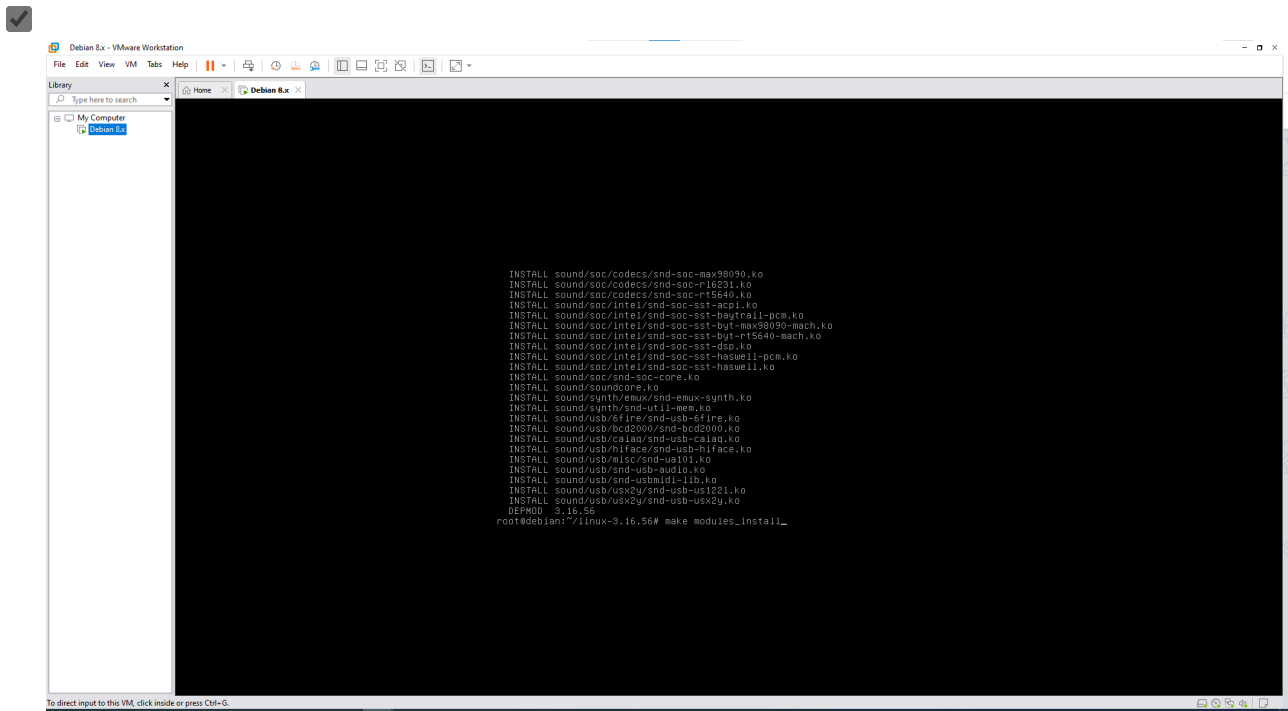
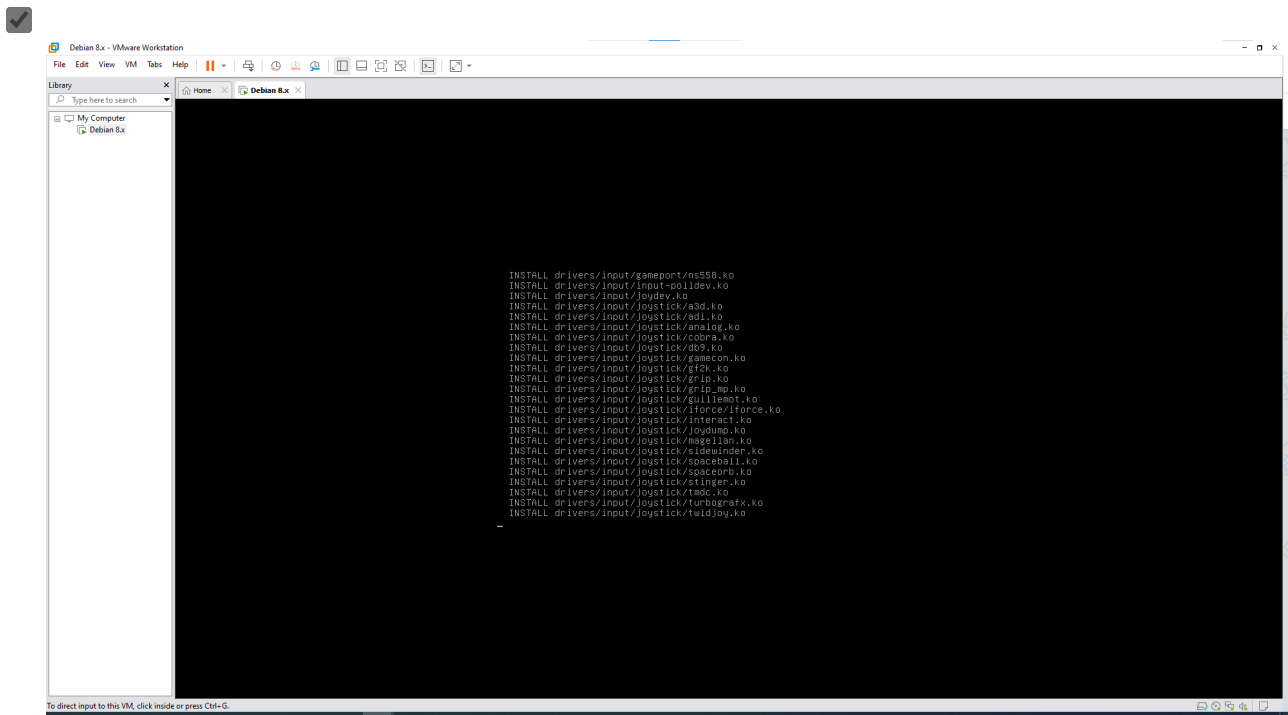
Added prototypes for system calls to the file `syscall.h`.



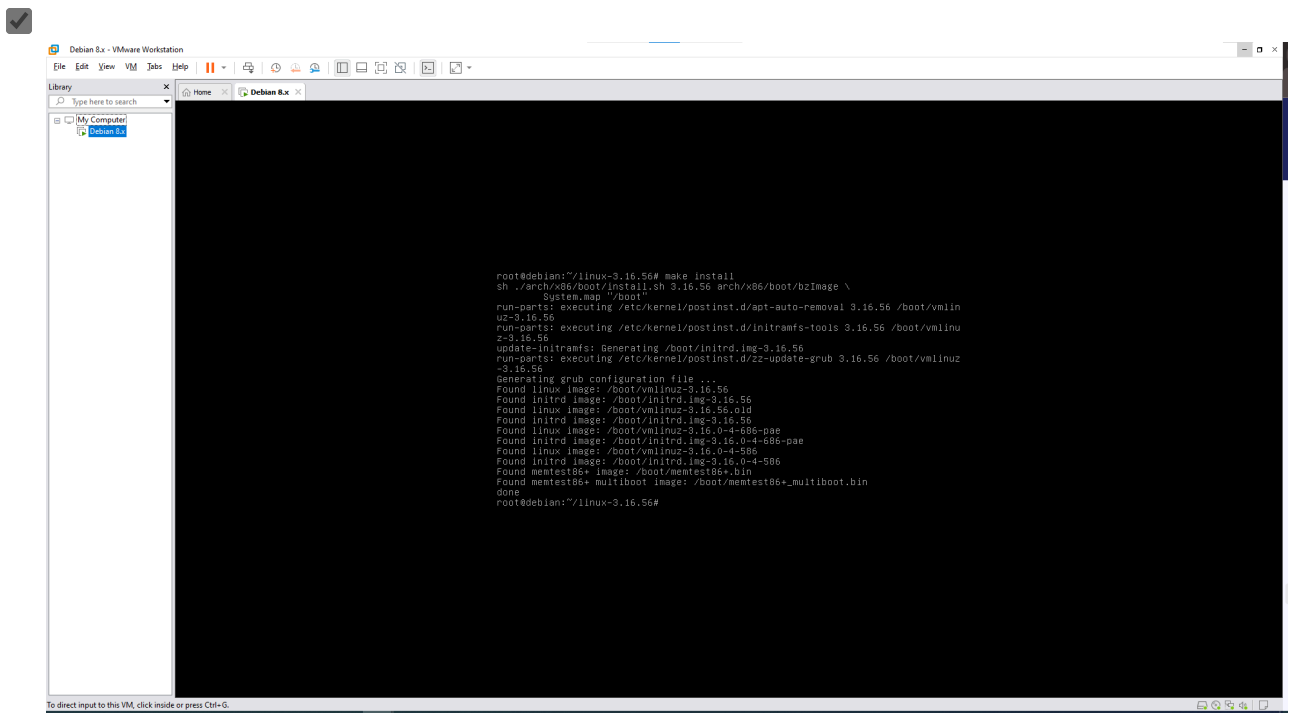
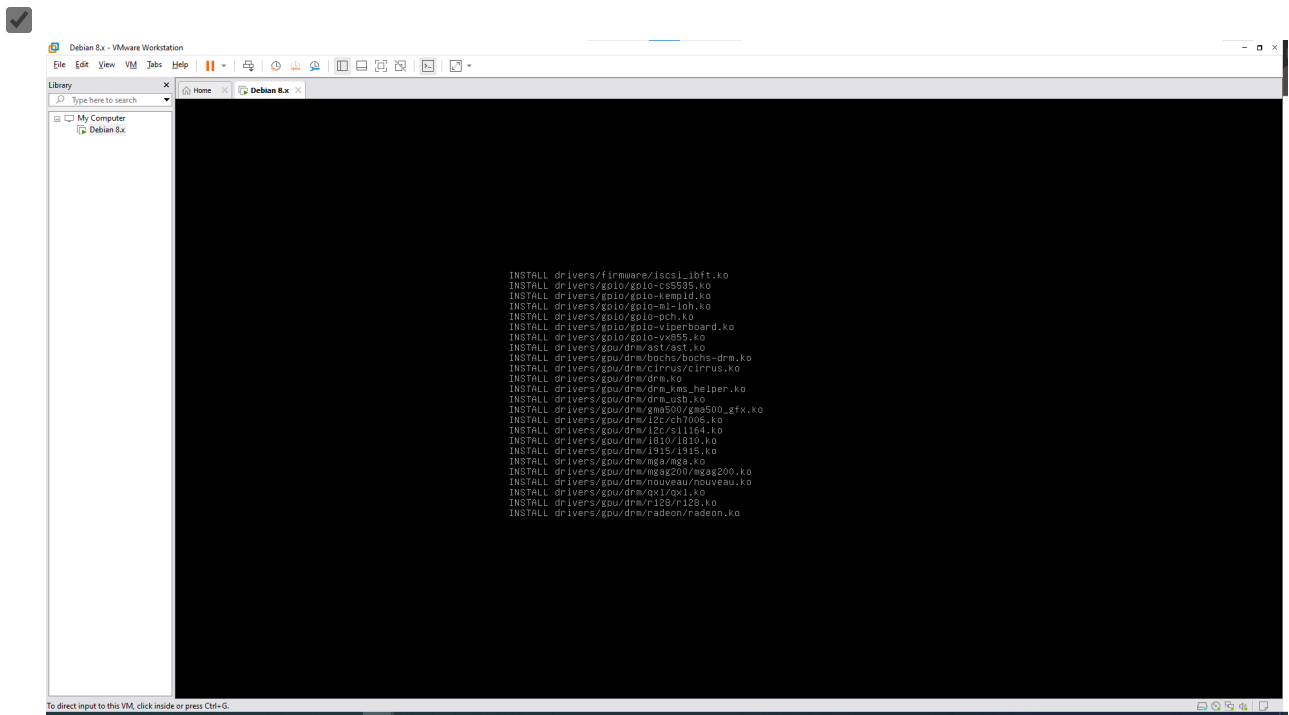
Run make oldconfig



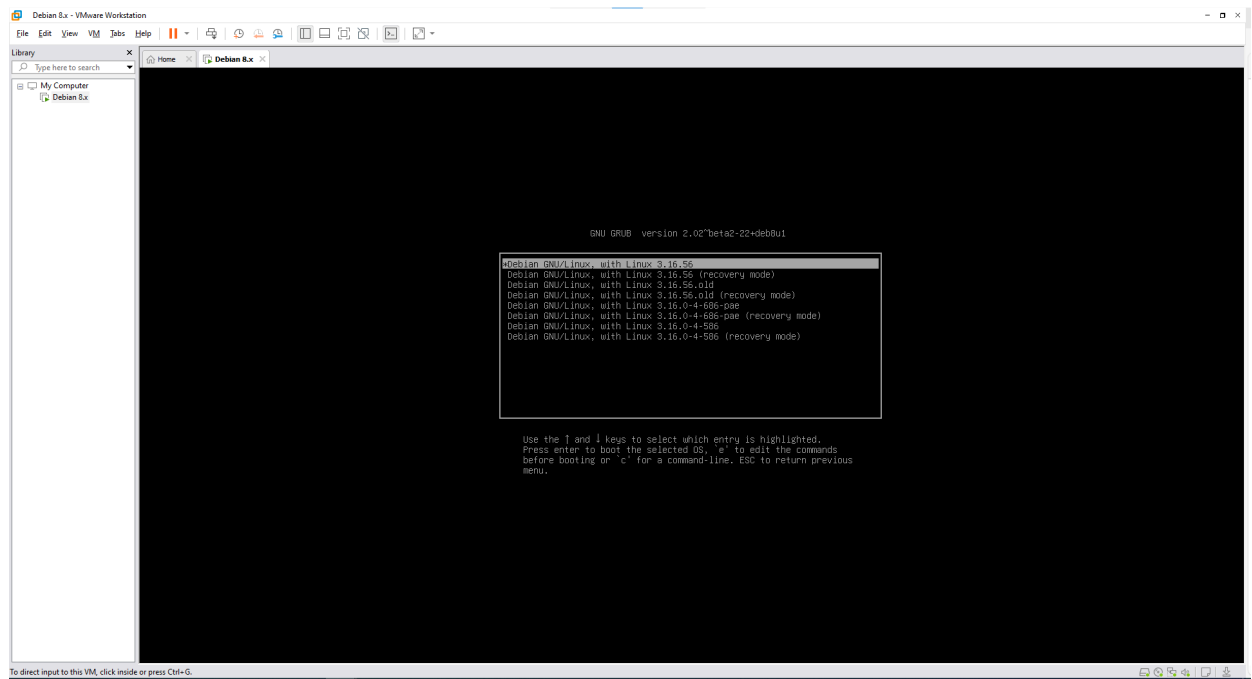
Run make



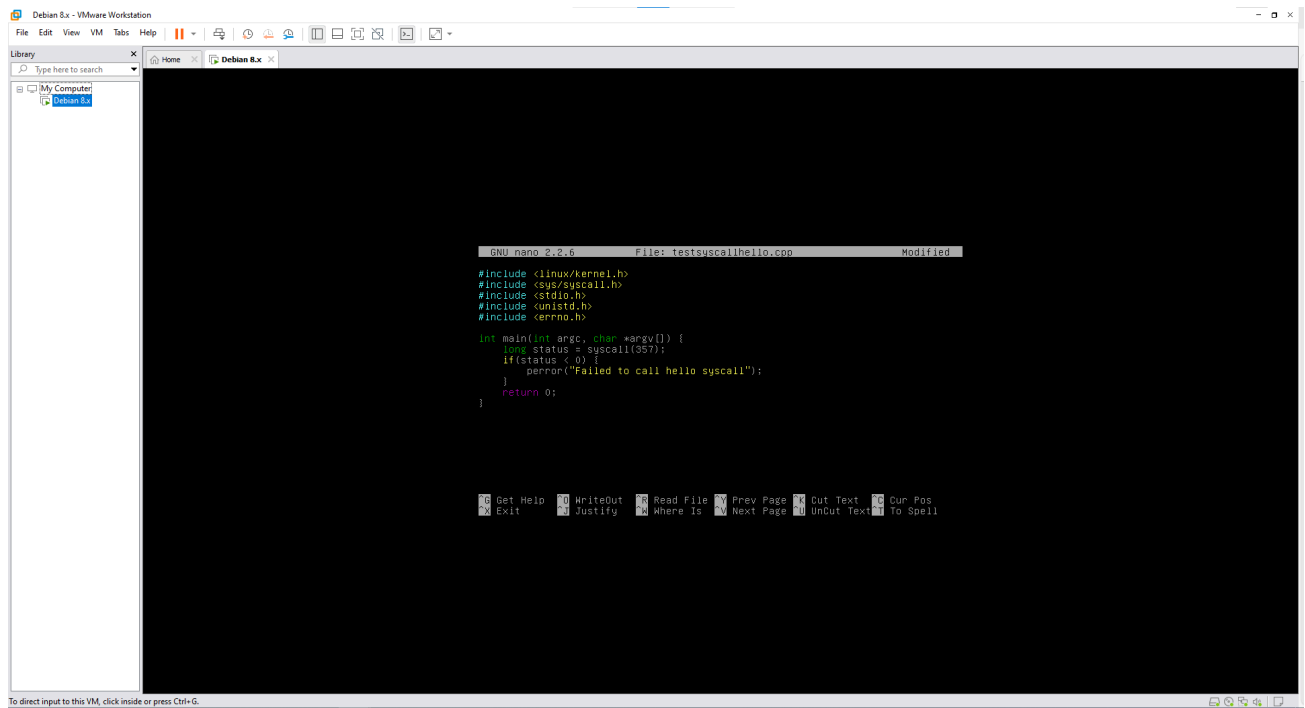
Run `make modules_install`



Run make install



Reboot system and choose compiled kernel





```
Debian GNU/Linux 8 debian tty1

debian login: root
Password:
Last login: Thu Jul 4 05:30:18 EDT 2024 on tty1
Linux debian 3.16.56 #4 SMP Thu Jul 4 05:43:42 EDT 2024 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright*.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# ls
a.out linux-3.16.56 testdir
linux-3.16.56-1-deb01.debian.tar.xz testsyscall2.cpp
linux-3.16.56-1-deb01.dsc testsyscall.cpp
linux-3.16.56.orig.tar.xz testsyscallhello.cpp
root@debian:~# g++ testsyscallhello.cpp
root@debian:~# ./a.out
root@debian:~# dmesg | tail -n 1
[35.733266] Hello World
root@debian:~#
```

To direct input to this VM, click inside or press Ctrl+G.



```
GNU nano 2.2.6 File: testsyscalladd.cpp

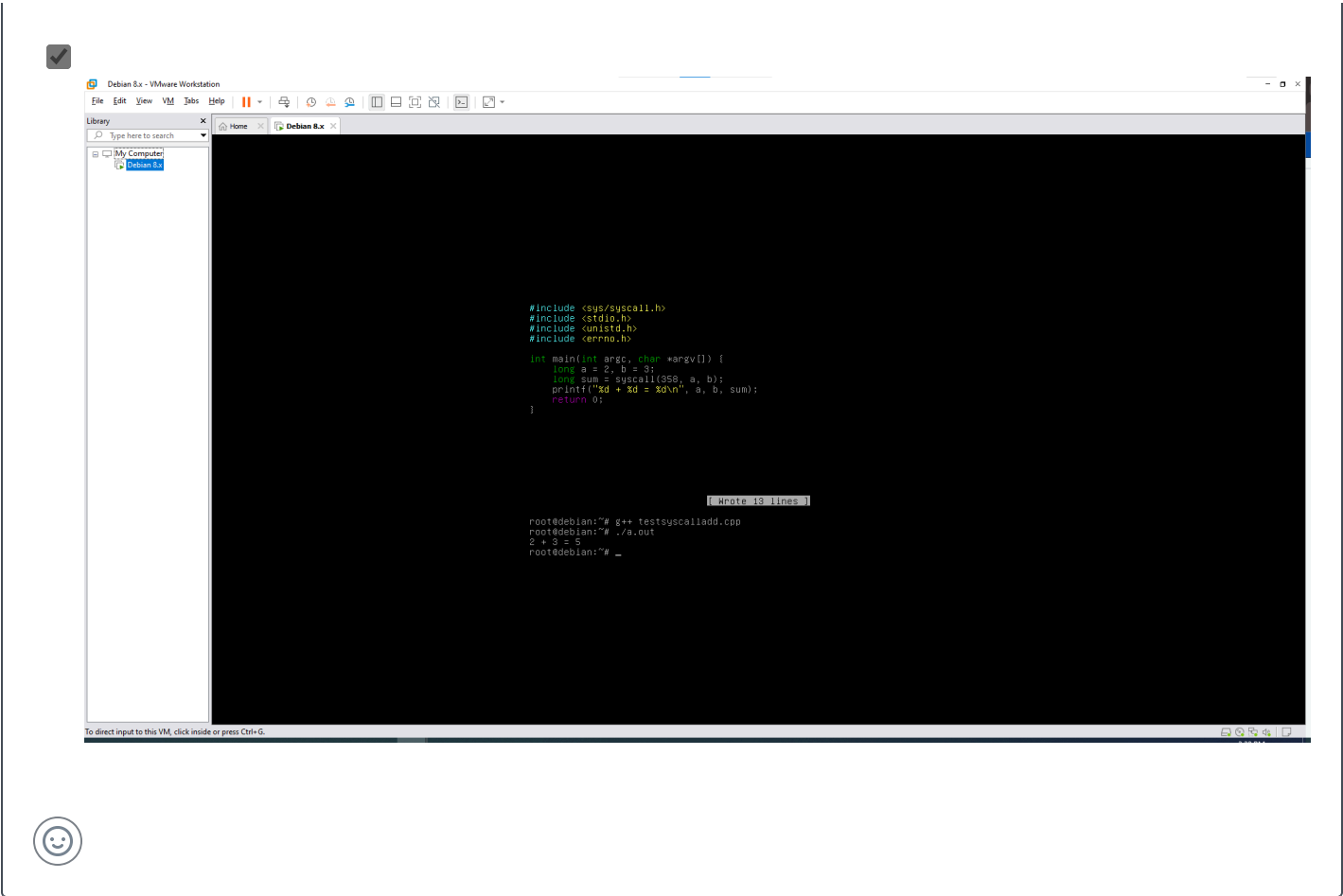
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <stdio.h>
#include <unistd.h>
#include <errno.h>

int main(int argc, char *argv[]) {
 long a = 2, b = 3;
 long sum = syscall(558, a, b);
 printf("%d + %d = %d\n", a, b, sum);
 return 0;
}

[Wrote 13 lines]

root@debian:~#
```

To direct input to this VM, click inside or press Ctrl+G.



BozorgmehrZia added the **documentation** label last week



BozorgmehrZia assigned Mahdiplj last week

## Assignees

 Mahdiplj

## Labels

**documentation**

## Projects

None yet

## Milestone

No milestone



---

# Development

No branches or pull requests