

 YYYYYYY · Follow
Last edited by a3020008 on Oct 4, 2023
Contributed by  

♡ 2

🔖

🔔

💬 0

- Assignment 1: Compiling...
- Appendix
- Expand all
- Back to top
- Go to bottom

Assignment 1: Compiling Linux Kernel and Adding Custom System Calls

- Assignment 1: Compiling Linux Kernel and Adding Custom System Calls
 - Compiling Linux Kernel
 - Change kernel suffix
 - Adding custom system calls
 - sys_hello
 - sys_revstr
 - Submission
- Appendix
 - Error: No space left on device

Summary: This assignment requires you to compile the linux kernel from source and define custom system calls.

Compiling Linux Kernel

In this assignment, you will need to compile the linux kernel on **Ubuntu 22.04 AMD64**. The Ubuntu image can be downloaded [here](#). There are desktop and server versions. Choose whichever you prefer.

For Mac M1/M2 users, you may work on **Ubuntu 22.04 ARM64**.

If you have no experience on installing linux before, you may want to check [here](#) to see how to run Ubuntu on a VM using VirtualBox.

Tips: The minimal disk size should be at least **50GB**, or you may encounter the error like "No space left on the device" when building your kernel.
We also recommend the number of CPU cores should be at least 4, or you may spend lots of time compiling the kernel.

The required kernel source version is **5.19.12**. You can directly download the source tarball from <https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.12.tar.xz>.

We will not go through the kernel compilation steps here and **also NOT be open to any questions you meet on building Linux kernel**, since there are tons of documents online.

Change kernel suffix

Before compiling the kernel, you need to change your Linux kernel version suffix to `—os—<your—id>` , as an evidence that the kernel is built by yourself.

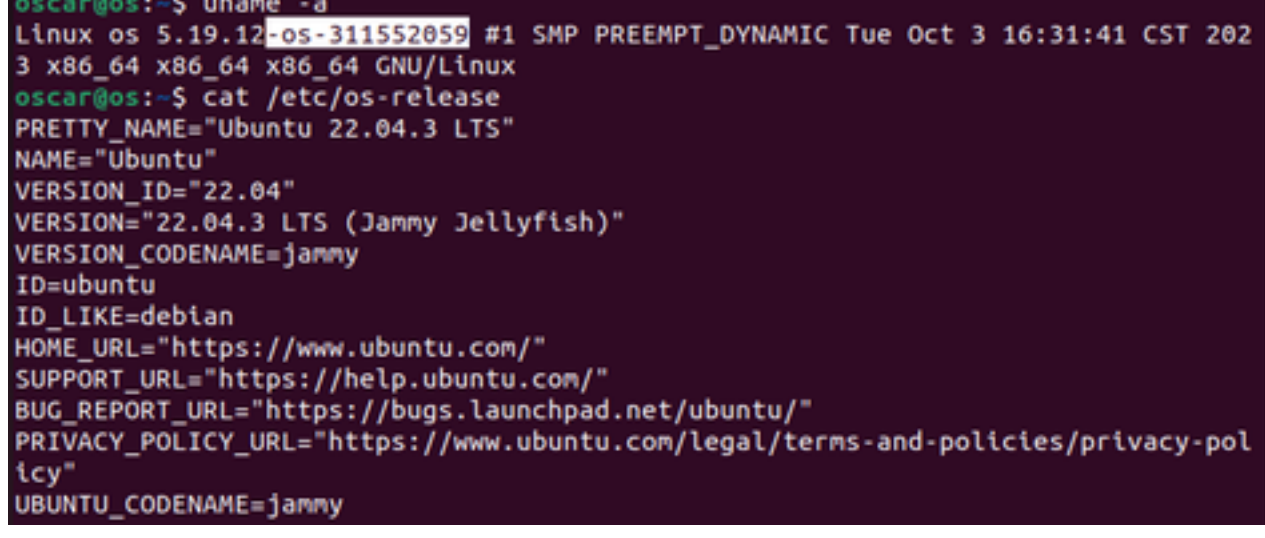
Hint💡: you may want to see `CONFIG_LOCALVERSION` config.

Here are some useful sites you may want to follow:

- <https://kernelnewbies.org/KernelBuild>
- https://wiki.archlinux.org/title/Kernel/Traditional_compilation

Example Screenshot

An example of a successful kernel compilation by showing the results of `uname -a` and `cat /etc/os-release` commands:



Adding Custom System Calls

In this part, you are required to implement two system calls on top of the kernel you built. One `sys_hello` and the other one `sys_revstr`.

You may want to check for <https://www.kernel.org/doc/html/latest/process/adding-syscalls.html#generic-system-call-implementation>

sys_hello

You are required to implement a system call called `sys_hello` , which does the following:

- Prints a line showing a string "Hello, world!".
- Prints another line showing a string "<your-id>", where `<your-id>` is your student id.
- Finally, returns zero indicating the system call was executed successfully.

The following code invokes the system call `sys_hello` and examines if the return value is zero. Please make sure the code can be compiled on your system and works correctly.

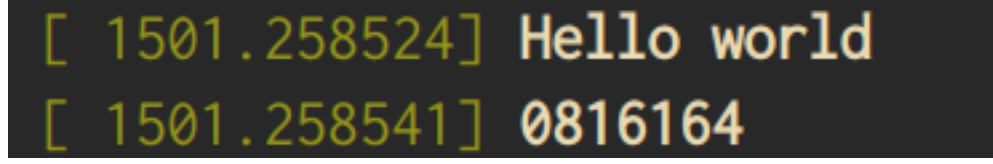
```
#include <assert.h>
#include <unistd.h>
#include <sys/syscall.h>

/*
 * You must copy the __NR_hello marco from
 * <your-kernel-build-dir>/arch/x86/include/generated/uapi/asam/unistd_64.h
 * In this example, the value of __NR_hello is 548
 */
#define __NR_hello 548

int main(int argc, char *argv[]) {
    int ret = syscall(__NR_hello);
    assert(ret == 0);

    return 0;
}
```

Besides, the kernel ring buffer should contain the messages that `sys_hello` printed. You can display them with `dmesg` command:



sys_revstr

This time, you are required to implement a system call called `sys_revstr` . It receives two arguments: the string length and a string, and then reverses the string.

Following are the steps you need to follow:

- First print a line "The origin string: <string>", where `<string>` is the string the user passed in.
- Reverse the given string.
- Print a line "The reversed string: <rev_string>", where `<rev_string>` is the resulting string you reversed.
- Finally, return zero indicating the system call was executed successfully.

Hint1💡: The data from the user space are not shared with the kernel space (and vice versa), so you need to figure out how to transfer data between them.

Hint2💡: You don't need to handle invalid arguments such as a null pointer or mismatched string length.

Please make sure the following code can be compiled on your system and works correctly.

```
#include <assert.h>
#include <unistd.h>
#include <sys/syscall.h>

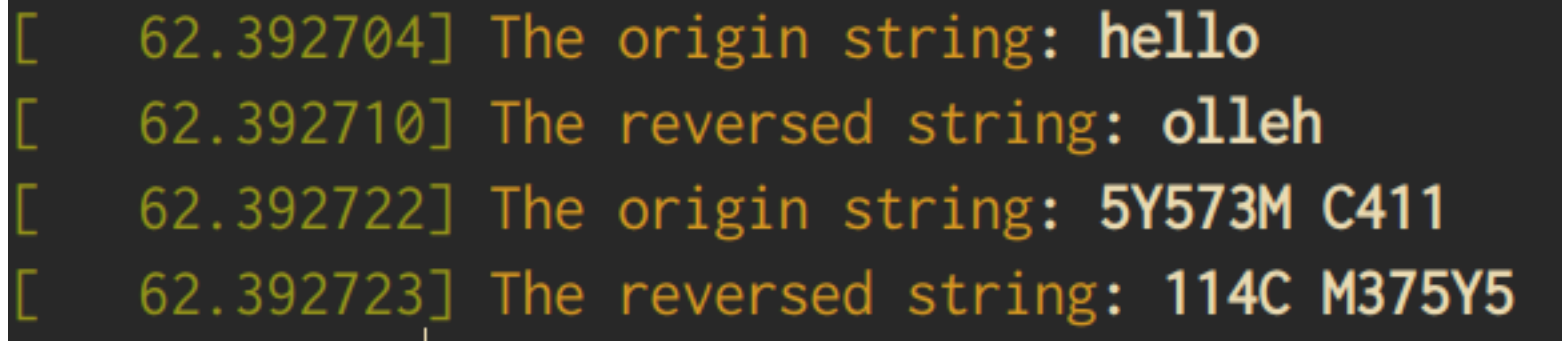
/*
 * You must copy the __NR_revstr marco from
 * <your-kernel-build-dir>/arch/x86/include/generated/uapi/asam/unistd_64.h
 * In this example, the value of __NR_revstr is 549
 */
#define __NR_revstr 549

int main(int argc, char *argv[]) {
    int ret1 = syscall(__NR_revstr, 5, "hello");
    assert(ret1 == 0);

    int ret2 = syscall(__NR_revstr, 11, "5Y573M C411");
    assert(ret2 == 0);

    return 0;
}
```

Finally, use `dmesg` to see the output messages:



Submission

Please submit a single **pdf** file to E3, which contains two parts: kernel compilation and system call.

For the kernel compilation part (4 pts):

- Paste the screenshot of the results of executing `uname -a` and `cat /etc/os-release` commands as the example shows.

For the system call part (3 pts for each system call):

- Describe how you implemented the two system calls in detail.
 - Which kernel sources did you modify? What do they do?
- For each system call you implemented:
 - Paste the **well-formatted** source code into the document.
 - Paste the screenshot of the messages the system call printed.

The filename should be `<id>_hw1.pdf` .

The deadline is on **10/31 23:59**.

Appendix

Error: No space left on device

By default, the Ubuntu installer allocates the logical volume of the root filesystem (`/dev/ubuntu-vg/ubuntu-lv`) for only half of space of the underlying physical volume `ubuntu-vg` .

To address this issue, you can extend the default LVM to 100%.

```
sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv
sudo resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv
```

Please refer to <https://packetpushers.net/ubuntu-extend-your-default-lvm-space/> for more information.

Last changed by 

 YYYYYYY · Follow

💬 0

♡ 2

👁 3070

🔖