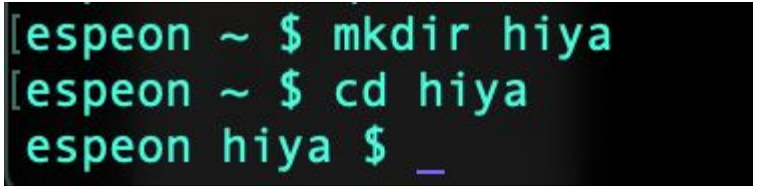
Operating Systems Final project

Qiyue Aixinjueluo
Jenna Berlinberg
Alvin Lai

Our Idea

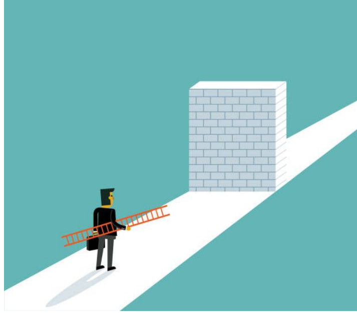
Normally:



```
[espeon ~ $ mkdir hiya
[espeon ~ $ cd hiya
 espeon hiya $ _
```

Why not combine them? So now, you will be able to jump into the new folder in only one line!

Obstacle 1



At first, we wanted to make a flag ‘mkdir -c <directory name>’.
This would involve modifying the mkdir system call.

Upon further research, we found that mkdir is not in the linux kernel. It is in the GNU Standard Library coreutils, which comes with any Ubuntu OS. Modifying it is highly risky and is not advised.

We decided to play it safe and just make a custom system call.

mkcd system call

We created the system call mkcd to combine the two steps.

Created using Ubuntu 19.10, Linux kernel 5.6.7, gave access to 4 cores so it compiles faster!

Started following the basic ‘create a hello world syscall’ tutorials in terms of setting it up, but instead of printing something, we needed to call two pre-existing system calls.

Obstacle 2

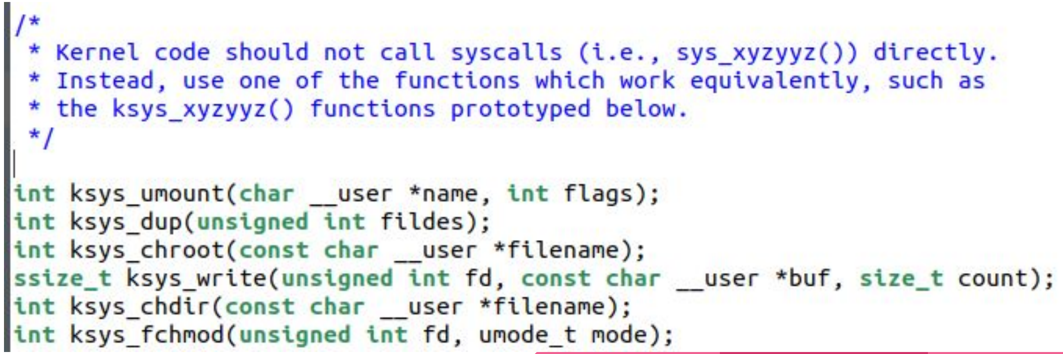
Calling system calls in kernel code was difficult to figure out.

First tried using the mkdir() and chdir() functions from C. Got the error functions not defined despite including proper files.

Reached out to Adjunct Professor at USF, former senior director at Motorola, Apple, etc, and Jenna's mom's tennis partner, Phil Peterson for help on digging around the Linux kernel.

His first insight was to call the sys_chdir and sys_mkdir functions. Tried those, but got the same error!

With some further sleuthing through the kernel code, we found the following goldmine in include/linux/syscalls.h line 1231:



```
/*
 * Kernel code should not call syscalls (i.e., sys_xyzyyz()) directly.
 * Instead, use one of the functions which work equivalently, such as
 * the ksys_xyzyyz() functions prototyped below.
 */

int ksys_umount(char __user *name, int flags);
int ksys_dup(unsigned int fildes);
int ksys_chroot(const char __user *filename);
ssize_t ksys_write(unsigned int fd, const char __user *buf, size_t count);
int ksys_chdir(const char __user *filename);
int ksys_fchmod(unsigned int fd, umode_t mode);
```

SHE LIVES!

One small final error- got undefined function error from the syscall table file, found an error in naming __x64_sys_mkcd via: `objdump -S mkcd.o` The function compiled to __x64_sys_sys_mkcd ... OTL

After fixing that, the kernel with the system call has **compiled!**

The fatal flaw is the fact that **you cannot make changes to the user location within a C file.** The system call itself will do what it is meant to- but only inside the child thread of the main() function on userspace.c. To officially make this callable in the command line, we'd need to make a command line call that would call the mkcd system call. Perhaps, a side project…?

DEMO

Thank you!

Questions?