

In-class: xv6 system calls

Part One: System call tracing

Your first task is to modify the xv6 kernel to print out a line for each system call invocation. It is enough to print the name of the system call and the return value; you don't need to print the system call arguments.

When you're done, you should see output like this when booting xv6:

```
...
fork -> 2
exec -> 0
open -> 3
close -> 0
$write -> 1
write -> 1
```

That's init forking and execing sh, sh making sure only two file descriptors are open, and sh writing the \$ prompt.

Hint: modify the syscall() function in syscall.c.

Challenge: print the system call arguments.

Part Two: Halt system call

Your second task is to add a new system call to xv6. The main point of the exercise is for you to see some of the different pieces of the system call machinery.

Your new system call will halt xv6 by telling QEMU to exit. Here is some code that does that; it needs to run in the kernel:

```
char *p = "Shutdown";
for( ; *p; p++)
    outb(0x8900, *p);
```

You should create a user-level program that calls your new halt system call; here's some source you should put in halt.c:

```
#include "types.h"
#include "stat.h"
#include "user.h"

int
main(int argc, char *argv[])
{
    halt();
    return 0;
}
```

In order to make your new halt program available to run from the xv6 shell, add `_halt` to the `UPROGS` definition in Makefile.

Your strategy for making a halt system call should be to clone all of the pieces of code that are specific to some existing system call, for example the "uptime" system call. You should grep for uptime in all the source files, using `grep -n uptime *.chS`.

When you're done, typing `halt` to an xv6 shell prompt should cause qemu to exit.

Write down a few words of explanation for each of the files you had to modify in the process of creating your halt

system call.

Challenge: add a dup2() system call.

Submit: Your explanations of the modifications for halt.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.828 Operating System Engineering
Fall 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.