xv6

hands-on session

CS347m/CS219
Autumn 2025

xv6 inside-out

`https://www.cse.iitb.ac.in/~puru/xv6.html`

A comprehensive introduction to xv6 features and exercises

Setup

Login using **labuser** as username and password

Download/copy the xv6 folder

```
curl https://www.cse.iitb.ac.in/~puru/courses/xv6-public.tar.gz --output xv6.tar.gz

tar -xvzf xv6.tar.gz

cd xv6-public

make qemu-nox

ls, echo, cat, wc

$ echo “hello world!” > hw.txt
$ cat hw.txt
$ wc hw.txt
```

Ctrl-a x (to quit)

xv6 system calls

Look for following files in the xv6 source code —

```
user.h
usys.S
syscall.h
syscall.c
sysproc.c
proc.c
```

Trace the **`fork`**, **`exit`**, **`exec`** system calls in each of these files

Change a system call implementation!

Inside the fork system call
add a line of code that prints the PIDs of process calling fork
and the newly created child process

```
cprintf, sys_fork, fork, struct proc …
sysproc.c, proc.c, proc.h …
```

```
$ ls
Fork called: 2 3
```

Adding a new user space program to xv6

Create a new user space program **`hw.c`**
which prints “Hello world!” to the console/terminal

Our task is to put the file and its compiled userspace program
inside the xv6 file system

Should be able to boot into xv6, find the **`hw.c`**, display the file and run the executable **`hw`**

To get started look into **`user.h, types.h, wc.c`**
and how the program **`wc`** is added to xv6

hw.c

```
$ cat hw.c
#include "types.h"
#include "user.h"

int main() {
      printf(1, "Hello World\n");
      exit();
}
```

Add a new system call

`getva2pa` —
takes an address (integer) as a single input argument and returns integer

`getva2pa`
needs a user call and system call implementation

Inside the call —
Find PCB, find pgdir, walkpagetable

Lookup `walkpgdir` function in xv6 kernel