xv6: Adding a user program

1. We want to run the following user program in xv6 (i.e. we want to execute it from xv6 shell):

#include <stdio.h>

int main()

{

printf("hello world\n");

int num;

scanf("%d", &num);

printf("%d^2 = %d\n", num, num \* num);

return 0;

}

We cannot directly write a program like this in xv6 because

* xv6 does not have the stdio.h library. All the library functions are declared in user/user.h.
* user/user.h does not have **scanf**.

For xv6, add the program becomes:

#include "kernel/types.h"

#include "kernel/stat.h"

#include "user/user.h"

int main()

{

printf("hello world\n");

char buf[10];

gets(buf, 9);

int num = atoi(buf);

printf("%d^2 = %d\n", num, num \* num);

return 0;

}

Here, we can see two other files being imported: kernel/types.h and kernel/stat.h. This is because user/user.h depends on them.

Also, **scanf** is simulated using **gets** and **atoi**.

1. Now, save this code as myprog.c inside the folder user.

xv6 does not have a gcc compiler inside. The code needs to be precompiled into the OS image. For this, edit the **UPROGS** variable in Makefileas follows.

UPROGS=\

$U/\_cat\

$U/\_echo\

$U/\_forktest\

$U/\_grep\

$U/\_init\

$U/\_kill\

$U/\_ln\

$U/\_ls\

$U/\_mkdir\

$U/\_myprog\ # add this line

$U/\_rm\

$U/\_sh\

$U/\_stressfs\

$U/\_usertests\

$U/\_grind\

$U/\_wc\

$U/\_zombie\

1. Compile and run

$ make clean; make qemu

1. Call the command **myprog** in shell and you will get the following output.

$ myprog

hello world

3

3^2 = 9

$

# Practice

Generate a patch file for this user program. Cleanup everything and restore xv6 to initial state. Apply the generated patch and run again.