Xv6 Operating System -add a user program

Difficulty Level : Medium • Last Updated : 14 Aug, 2020

Xv6 is a re-implementation of the Unix sixth edition in order to use as a learning tool. xv6 was developed by MIT as a teaching operating system for their "6.828" course. A vital fact about xv6 is that it contains all the core Unix concepts and has a similar structure to Unix even though it lacks some functionality that you would expect from a modern operating system. This is a lightweight operating system where the time to compile is very low and it also allow remote debugging.

The source code of xv6 can be cloned to your machine as follows :
https://github.com/mit-pdos/xv6-public

Simple xv6 program

dding user program to xv6 :

fter completing xv6 setup on your machine, you could have a look at how to add a new user program to xv6. A user program cou a simple C program. However, just

adding a file to the xv6 folder would not be sufficient as we need to make it available to the user at the shell prompt.

Step-1: A simple C program

First of all, let's create a C program like the following. We save it inside the source code directory of xv6 operating system with the name first.c or whatever the name you prefer.

C

```
//A Simple C program
include "types.h"
include "stat.h"
include "user.h"

 //passing command line arguments

int main(int argc, char *argv[])
{
  printf(1, "My first xv6 program learnt at GFG\n");
  exit();
}

// This code is contributed by sambhav228
```

Step-2: Edit the Makefile

The Makefile needs to be edited to make our program available for the xv6 source code for compilation.

```
gedit Makefile
```

This line of code would open the Makefile in the gedit text editor. Next, the following sections of the Makefile needs to be edited to add your program first.c

```
UPROGS=\
_cat\
_crash\
_echo\
_factor\
_forktest\
_grep\
```

```
_hello\
_init\
```

Related Articles

```
_null\
_rm\
_sh\
_share\
_stressfs\
_usertests\
_wc\
_zombie\
_first\
EXTRA=\

 mkfs.c ulib.c user.h cat.c echo.c forktest.c grep.c kill.c\
 ln.c ls.c mkdir.c rm.c stressfs.c usertests.c wc.c zombie.c\
 first.c\
 printf.c umalloc.c\
 README dot-bochsrc *.pl toc.* runoff runoff1 runoff.list\
 .gdbinit.tmpl gdbutil\
```

Now, our Makefile and our user program is ready to be tested. Enter the following commands to compile the whole system.

```
make clean
make
```

Now, start xv6 system on QEMU and when it booted up, run ls command to check whether our program is available for the user. If yes, give the name of that executable program, which is in my case is first to see the program output on the terminal.

Output :

"My first xv6 program learnt at GFG" shows this output on the QEMU emulator window.

Previous Next

RECOMMENDED ARTICLES

Page : 1 2 3

Article Contributed By :

sambhav228
@sambhav228

Vote for difficulty

Current difficulty : Medium

Easy | Normal | Medium | Hard | Expert

Article Tags : Operating Systems

Practice Tags : Operating Systems

Improve Article | Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org