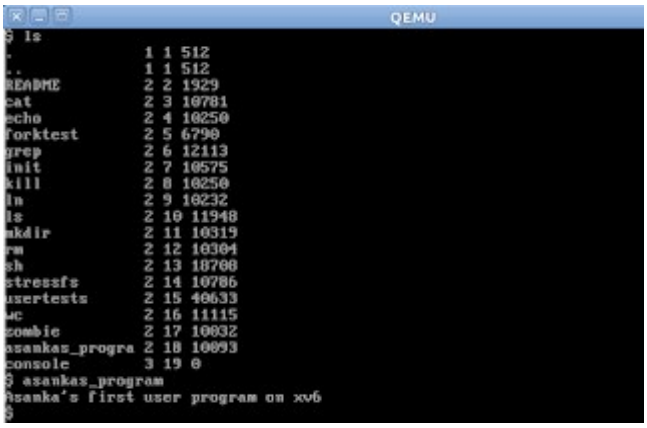
Adding a User Program to xv6

Link :
https://recolog.blogspot.com/2016/03/adding-user-program-to-xv6.html



It is pretty much simple and straightforward to write a user level C program for xv6 and making it available for the user at the shell prompt. When we go forward and implement much advanced things on the xv6 OS, most of the times we need to test those features by using a user level program. Therefore, knowing how to write such programs and adding it into the xv6 source code for compilation is a necessary thing. Therefore, I decided to write down those simple steps even though it is very simple.

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 **`asankas_program.c`** or whatever the name you prefer.

```
1 // A simple program which just prints something on screen
2
3 #include "types.h"
4 #include "stat.h"
5 #include "user.h"
6
7 int
8 main(void)
9 {
10  printf(1, "Asanka's first user program on xv6\n");
11  exit();
12 }
```

Now, we have to edit the **`Makefile`** of the xv6 source code in order to let the compiler know that we have a program like this and we need it to be compiled with other system programs. In the **`Makefile`**, there are two places in which we need to put entries.

Find the place with some lines like the following. We have to add a line as shown below to notify about our new program.

```
UPROGS=\
    _cat\
    _echo\
    _forktest\
    _grep\
    _init\
    _kill\
    _ln\
    _ls\
    _mkdir\
    _rm\
    _sh\
    _stressfs\
    _usertests\
    _wc\
    _zombie\
    _asankas_program\
```

Similarly, find the place with the lines like below. Add an entry as shown to indicate that we have a program called **asankas_program.c** there.

```
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\
    asankas_program.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 **asankas_program** to see the program output on the terminal. The figure above shows this output on the QEMU emulator window.