1. When xv6 boots up, firstly init process gets started. This is the first user process.

2. Then init process forks the shell process, and this shell is the main screen which is visible to us when we boot xv6.

3. In the shell we can give input commands, and these commands are executed by the shell.

4. After giving input command, the shell executes that command in the following steps:

   a. Firstly shell reads the command from the xv6 terminal by calling the read system call for each character in the input and stores it in a buffer, and based on the input command shell invokes the required system call.

   b. Then shell forks child and a new process is created.

   c. Then this child process runs by calling exec system call.

   d. Then shell(the parent) waits for the child process to complete, I.e., to terminate and then shell reaps it. And then

   e. Shell goes back to the prompt again.

5. And this whole process gets repeated.

This way the xv6 terminal runs user commands.

But its not always necessary that shell forks child process, because there are few commands which have to be executed by the parent process itself.

Example: cd command has to run in the terminal itself, and not by the child.

All the system calls that are available are defined in user.h file and all system call have unique numbers, these are defined in syscall.h file.

On calling a system call, it invokes a trap instruction in usys.S file using the int instruction.

Trap (int) instruction – makes a jump from user to kernel code and the functions related to system call are invoked.

As mentioned earlier, each system call has a number, and this number is moved into a register so that operating system can identify and exec the particular system call.

After executing the syscall, the return value of the system call, that is the result is stored in eax register.

The implementation of process related syscalls is done in sysproc.c.

Then after exec, the process cleans up the state, makes itself a zombie process and then it invokes the scheduler, claimimg I am done, and exits.