```
/**************************
Programer: Michael Marelli
Class: CS460
Project: Final
Date: 12/9/2016
File: login.c
***************************/

//*************************************************************************
//                LOGIC of login.c file
//*************************************************************************
char *tty;

#include "ucode.c"

main(int argc, char *argv[])   // invoked by exec("login /dev/ttyxx")
{
 char myname[16], passwd[16], line[256], buf[1024];
 char* token;
 int passfd, i, j, verified, uid, gid, end;
 tty =  argv[1];
 //1.
 close(0); close(1); close(2); // login process may run on different terms

 //2. // open its own tty as stdin, stdout, stderr
 open(argv[1], O_RDONLY); open(argv[1], O_WRONLY); open(argv[1], O_WRONLY);
 //3.
 settty(tty);   // store tty string in PROC.tty[] for putc()

  // NOW we can use printf, which calls putc() to our tty
  printf("MIKELOGIN : open %s as stdin, stdout, stderr\n", tty);

  signal(2,1);  // ignore Control-C interrupts so that
              // Control-C KILLs other procs on this tty but not the main sh

  while(1){
    verified = 0;
    i = 0;
    end = 0;
   // 1. show login:          to stdout
    printf("Login:");
    //2. read user nmae       from stdin
    gets(myname);
   // 3. show passwd:
    printf("Password:");
   // 4. read user passwd
    gets(passwd);
   // 5. verify user name and passwd from /etc/passwd file
    passfd = open("/etc/passwd", O_RDONLY);
    read(passfd, buf, 1024);
    while(!end)
    {
      //get new line
      bzero(line, 256);
      j = 0;
      while(buf[i] && buf[i] != EOF && buf[i] != '\n')
      {
        line[j++] = buf[i++];
        if(i >= 1024)
        {
```

```
        bzero(buf, 1024);
        if(!read(passfd, buf, 1024))
        {
          end = 1; //nothing more in the file to read
        }
        i = 0;
      }
    }
    if(!buf[i] || buf[i] == EOF)
    {
      end = 1;
    }
    i++; //we want to move past the newline for next time

    //evaluate line
    token = strtok(line, ":");
    if(!strcmp(token, myname)) //username found
    {
      token = strtok(0, ":"); //token is user passwd
      if(!strcmp(token, passwd))
      {
        verified = 1;
      }
      break;
    }
  }
  close(passfd); //we've already found the right line. we can close the file
  //6. if (user account valid){
  if(verified)
  {
    printf("LOGIN : Verified\n");
    //    setuid to user uid.
    token = strtok(0, ":"); //group id
    gid = atoi(token);
    token = strtok(0, ":"); //user id
    uid = atoi(token);
    chuid(uid, gid);

    //  chdir to user HOME directory.
    token = strtok(0, ":"); //full name
    token = strtok(0, ":"); //home dir
    chdir(token);

    //exec to the program in users's account
    token = strtok(0, ":"); //program to run
    exec(token); //process runs the user program will return
  }
  else
  {
    printf("login failed, try again\n");
  }
 }
}
```