```c
/****************** login.c ******************/
// login.c : Upon entry, argv[0]=login, argv[1]=/dev/ttyX
#include "ucode.c"
int in, out, err;
char user_name[128],password[128];
char *temp[8];

int main(int argc, char *argv[])
{
        close(0);close(1);//(1). close file descriptors 0,1 inherited from INIT.
        //(2). open argv[1] 3 times as in(0), out(1), err(2).
        in = open(argv[1],O_RDONLY);
        out = open(argv[1],O_WRONLY); //display to console
        err = open(argv[1],O_WRONLY); // display to console

        int gid,uid;

        settty(argv[1]); // set tty name string in PROC.tty
        //(4). open /etc/passwd file for READ;
        int passwd_fd = open("/etc/passwd",O_RDONLY); //password file descriptor
        while(1)
        {
                printf("Login>"); gets(user_name);
                user_name[strlen(user_name)] = 0;//kill the '\n'
                printf("Enter password>"); gets(password);
                password[strlen(password)] = 0;//kill the '\n'
                //printf("username=%s,password=%s\n",user_name,password);
                //for each line in /etc/passwd file do
                char *token;
                char line[100];
                int n = 0;
                while(n = readline(passwd_fd,line))
                {
                        //parse the line
                        parsePassword(line);
                        //printf("temp[0]=%s,temp[1]=%s\n",temp[0],temp[1]);
                        if(strcmp(user_name,temp[0]) == 0 && strcmp(password,temp[1])
== 0)
                        {
                                //login successful
                                gid = atoi(temp[2]);
                                uid = atoi(temp[3]);
                                prints("login is successful\n");
                                printf("gid=%d,uid=%d\n",gid,uid);
                                chuid(gid,uid);
                                if(strcmp(user_name,"root") == 0)
                                {
                                        //for the root user
                                        chdir("/");
                                }
                                else
                                {
                                        char user_path[50] = "/user/";
                                        strcat(user_path,user_name);
                                        printf("userpath=%s\n",user_path);
                                        chdir(user_path);
                                }

                                close(passwd_fd);
                                printf("executing shell program\n");
                                exec("sh");
                        }
```

```
                }
                prints("login failed, try again\n");
                close(passwd_fd);
                passwd_fd = open("/etc/passwd",O_RDONLY);//open for read again
        }
}

int readline(int fd, char *buf)
{
        int i = 0;
        char temp[5] = "";
        strcpy(buf,"");//clear buf
        while(strcmp(temp,"\n") != 0){
                if(read(fd,temp,1) > 0)
                {
                        strcat(buf,temp);
                        i++;
                }
                else //if read(fd,temp,1) == 0
                        break;
        }
        return i;
}

int parsePassword(char *line)
{
  char *cp = line;
  argc = 0;
  while (*cp != 0){
    while (*cp == ':')  *cp++ = 0;
    if (*cp != 0)// skip over blanks // token start
      temp[argc++] = cp; // pointed by argv[ ]
    while (*cp != ':' && *cp != 0) // scan token chars
      cp++;
    if (*cp != 0)
      *cp = 0;
    else // end of token
      break; // end of line
    cp++;// continue scan
  } //end outer while
  temp[argc] = 0; // argv[argc]=0
}
```