

```

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

//*****
//          Logic of init.c
// NOTE: this init.c creates only ONE login process on console=/dev/tty0
// YOUR init.c must also create login processes on serial ports /dev/ttyS0
// and /dev/ttyS1..
//*****
#define TTY0 0
#define TTYS0 1
#define TTYS1 2

int pid = 0, dev1 = 0, dev2 = 0, dev3 = 0, status;
int stdin, stdout;

#include "ucode.c" //<===== AS POSTED on class website

main(int argc, char *argv[])
{
    //1. // open /dev/tty0 as 0 (READ) and 1 (WRITE) in order to display messages
    stdin = open("/dev/tty0", O_RDONLY);
    stdout = open("/dev/tty0", O_WRONLY);
    //2. // Now we can use printf, which calls putc(), which writes to stdout
    printf("MIKEINIT : fork login tty0 on console\n");
    dev1 = fork();

    if (dev1) //parent
    {
        printf("MIKEINIT : fork login ttyS0 on console\n");
        dev2 = fork();
        if (dev2) //parent
        {
            printf("MIKEINIT : fork login ttyS1 on console\n");
            dev3 = fork();
            if (dev3) //parent
            {
                parent();
            }
            else //Proc 4
            {
                login(TTYS1);
            }
        }
        else //Proc 3
        {
            login(TTYS0);
        }
    }
    else //Proc 2
    {
        login(TTY0);
    }
}

```

```
int login(int dev)
{
    switch(dev)
    {
        case TTY0:
            exec("login /dev/tty0");
            break;
        case TTYS0:
            exec("login /dev/ttyS0");
            break;
        case TTYS1:
            exec("login /dev/ttyS1");
            break;
    }
}

int parent()
{
    while(1){
        printf("MIKEINIT : waiting for login child to die.....\n");

        pid = wait(&status);

        if (pid == dev1)
        {
            dev1 = fork();
            if(dev1)
            {
                //do nothing parent continues to wait for children to die
            }
            else
            {
                login(TTY0);
            }
        }
        else if (pid == dev2)
        {
            dev2 = 0;
            dev2 = fork();
            if(dev2)
            {
                //do nothing parent continues to wait for children to die
            }
            else
            {
                login(TTYS0);
            }
        }
        else if (pid == dev3)
        {
            dev3 = 0;
            dev3 = fork();
            if(dev3)
            {
                //do nothing parent continues to wait for children to die
            }
            else
            {
                login(TTYS1);
            }
        }
    }
}
```

```
    }  
    else  
        printf("INIT : buried an orphan child %d\n", pid);  
    }  
}
```