Advanced Operating System Software Project

Gwan-Hwan Hwang
Dept. CSIE
National Taiwan Normal University
Goal of the project

- Let students practice programming for a multi-tasking software
- What can students learn from the implementation of this project?
  - Process management
  - Interprocess communication
    - Pipe
    - Message queue
  - Redirection of I/O
  - Screen layout in curses package
Platform

- Unix operating system with standard System V system calls
What to implement?

- A multi-window shell
  - Four windows for printing the executing results of the commands which are inputted by the user
    - W0, W1, W2, and W3
  - One window for the user to input his command
    - Command window
What to implement? (Cont’d)

```bash
$ who
root     tty1         Sep  6 08:08
u89405   pts/0        Nov 11 08:37 (140.122.65.54)
ghhwang  pts/2        Nov 11 10:38 (61-216-119-89.dynamic.hinet.net)
zordius  pts/1        Nov  8 21:06 (61-230-65-68.dynamic.hinet.net)
ghhwang@www ~ $ who
root     tty1         Sep  6 08:08
u89405   pts/0        Nov 11 08:37 (140.122.65.54)
ghhwang  pts/2        Nov 11 10:38 (61-216-119-89.dynamic.hinet.net)
zordius  pts/1        Nov  8 21:06 (61-230-65-68.dynamic.hinet.net)
ghhwang@www ~ $ who
root     tty1         Sep  6 08:08
u89405   pts/0        Nov 11 08:37 (140.122.65.54)
ghhwang  pts/2        Nov 11 10:38 (61-216-119-89.dynamic.hinet.net)
zordius  pts/1        Nov  8 21:06 (61-230-65-68.dynamic.hinet.net)
```

```bash
$ man:x:13:15:man:/usr/man:/bin/false
postmaster:x:14:12:postmaster:/var/spool/mail:/bin/false
cron:x:16:16:cron/var/spool/cron:/bin/false
tftp:x:21:21:/tftpboot/bin/false
sshd:x:22:22:sshd:/usr/sbin/sshd:/bin/false
at:x:25:25:at:/var/spool/at:/bin/false
squid:x:31:31:Squid/var/cache/squid:/bin/false
gdm:x:32:32:GDM/var/lib/gdm:/bin/false
gxfs:x:33:33:X Font Server/etc/X11/fs:/bin/false
games:x:35:35:games/usr/games:/bin/false
```
Syntax of command

- \texttt{W?://[shell command]}
  - \texttt{W0://ls -l}
  - \texttt{W1://cp temp1.c temp2.c}
  - \texttt{W3://ps aux|grep ghhwang}
Suggested system architecture

Pm

Pipe c0 → P0 → Pipe r0 → P0p
Pipe c1 → P1 → Pipe r1 → P1p
Pipe c2 → P2 → Pipe r2 → P2p
Pipe c3 → P3 → Pipe r3 → P3p

IPC_NOWAIT

Message queue M
Processes

- **Pm:**
  - To receive command from the user
  - To fork processes to execute shell commands
  - To print the execution result to the appropriate screen

- **P1, P2, P3, and P4:**
  - forked process to execute shell commands

- **P1p, P2p, P3p, and P4p:**
  - Processes which add prefixes to the execution results received from P1, P2, P3, and P4
Suggested system architecture (Cont’d)

- Pipes
  - Pipe c0, c1, c2, c3: transmit commands
  - Pipe r0, r1, r2, c3: transmit execution results

- Message queue M
  - Store the prefixed execution results
Message format of the prefixed execution results in message queue $M$

$W0/234/$

It represents the name of the window which the execution results should be shown.

Length of the execution results in bytes.
I/O system

- UNIX 螢幕導向程式的發展利器 – curses
  - Check the homepage of the teacher.
- manual page of curses
void initial()
{
    initscr();
    cbreak();
    nonl();
    noecho();
    intrflush(stdscr, FALSE);
    keypad(stdscr, TRUE);
    refresh();
}

#include <curses.h>
#include <sys/signal.h>

void initial()
{

    main()
    {
        WINDOW *win[3], *curwin, *helpwin;
        int nowwin;
        int x, y;
        int i, j;
        int ch;
        x = 2; y = 2;

        initial();
        nodelay(stdscr, TRUE);

        win[0] = newwin(LINES / 3 - 1, COLS - 1, 0, 0); /* 設定兩個視窗的大小*/
        win[1] = newwin(LINES / 3 - 1, COLS - 1, LINES / 3, 0);
        win[2] = newwin(4, COLS - 1, 20, 0);

        scrollok(win[0], TRUE);
        scrollok(win[1], TRUE);
        scrollok(win[2], TRUE);

        j = 1;
        for (;;)
        {
            j++;
            wprintw(win[0], "test%d for window1 \n", j);
            /* wrrefresh(win[0]);
            */
            curwin = win[2];
            wmove(curwin, y, x);
            if ((ch = getch()) != ERR) {
                waddch(win[2], ch);
                wrefresh(win[2]);
                x++;
            } else
            {
                if (j % 100 == 0) { wrefresh(win[0]); wrefresh(win[1]);}
                wprintw(win[1], "test%d for window2 \n", j);
                /* wrrefresh(win[1]);
                */
            }
        }
    }
}
Bonus points

- In case you can design your own architecture
- You can relinquish the use of the command window
  - For example, you can set up to hit the TAB key to switch between output windows W0, W1, W2, and W3 and input command in the selected output window.
- Your system can handle signal
- Others
  - Whatever you think is reasonable.
Project report

- The student should prepare a report which contains at least the follows:
  - The architecture of the implemented software
  - What you have learned and experienced during the implementation.
    - E.g., you could show your daily record of the implementation.
  - In case you implement more than the required specification, please itemize it.
- Copyright Claim
  - Do you make the implementation yourself?
- Any thing you would like to let G.H.Hwang know.
  - E.g. Suggestion, …

- Who will be reading the report?
  - Not TAs but G. H. Hwang
How to hand in your report?

- Please send a mail to TA with a zip file
  - Mail title: OS final project + your student id
    - E.g., “OS final project D828302”
  - Attached filename: your_student_id.zip
  - It should have at least the following items:
    - Electronic files of your report
      - PDF format
    - Source codes
      - A readme.txt to show how to compile your code