

[506489] 시스템프로그래밍

실습 #06 문제 및 보고서

이름	곽영주
학번	20175105
소속 학과/대학	빅데이터
분반	01 (담당교수: 김태운)

<주의사항>

- 개별 과제입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
- **각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.**
 - 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.
- **SmartLEAD 제출 데드라인:**
 - **월요일 분반: 다음 실습 시간 전날까지(일요일 까지)**
 - **수요일 분반: 다음 실습 시간 전날까지 (화요일 까지)**
 - 데드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점(5 일 경과 시 0 점)
 - 주말/휴일/학교행사 등 모든 날짜 카운트 함
 - 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함
 - 예외 없음
- **SmartLEAD 에 아래의 파일을 제출해 주세요**
 - **보고서(PDF 파일로 변환 후 제출 권장하나, WORD 형식으로 제출도 가능)**
 - 보고서 파일명에 이름과 학번을 입력해 주세요.
 - **소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (zip 파일로 압축하여 하나의 파일로 제출)**

<개요>

이번 과제는 시스템 정보 관리 및 프로세스 정보 관리에 대한 내용입니다.

<실습 과제>

[Q 0] 요약 [배점: 10]

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.

답변: 이번 과제를 통해 c 언어로 시스템 정보 관리 및 프로세스 정보 관리를 할 수 있다는 것을 알게 되었습니다. 그리고 ncurses 과제를 통해 소스코드를 조금씩 변경해보면서 ncurses 의 코드를 읽을 수 있게 되었습니다.

[Q 1] System Summary + 명령행 인자 [배점: 10]

강의노트의 [uname 함수 사용하기] 예제를 실행하면, 아래와 같은 결과를 출력합니다. 컴퓨터 시스템에 따라 출력되는 결과는 약간 다를 수 있습니다.

```
$ unameEx.out
OSname : Linux
Nodename : danpc1
Release : 4.15.0-62-generic
Version : #69-Ubuntu SMP Wed Sep 4 20:55:53 UTC 2019
Machine : x86_64
```

해당 예제 코드를 참고하여 아래와 같이 동작하는 systemSummary 프로그램을 작성하시오.

[사용 예 1] \$./systemSummary 입력 시 (즉, 아무런 인자 없이 프로그램을 실행하는 경우), 아래와 같이 도움말을 출력

```
systemSummary available options:
-s : print OS name
-n : print nodename (hostname)
-v : print release
-m : print machine info
```

[사용 예 2] \$./systemSummary -s 입력 시, 아래와 같이 출력

```
This is a Linux machine.
```

[사용 예 3] `./systemSummary -s -n` 입력 시, 아래와 같이 출력

```
This is a Linux machine named danpc1.
```

[사용 예 4] `./systemSummary -s -n -v` 입력 시, 아래와 같이 출력

```
This is a 4.15.0-62-generic Linux machine named danpc1.
```

[사용 예 5] `./systemSummary -s -n -v -m` 입력 시, 아래와 같이 출력

```
This is a x86_64 4.15.0-62-generic Linux machine named danpc1.
```

[사용 예 6] `./systemSummary -n -m` 입력 시, 아래와 같이 출력

```
This is a x86_64 machine named danpc1.
```

이와 같이, 어떠한 조합이라도 가능하며, 옵션이 주어진다면 해당하는 문구를 포함하도록 문장을 구성해야 함.

프로그램을 [사용 예 1~6]과 같이 구동하고, 터미널 입력 및 출력 화면을 캡처하시오. 캡처 이미지는 본 문서에 첨부하고, 소스코드는 첨부파일로 제출하시오.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w06/q01$ ./systemSummary
systemSummary available options:
-s : print OS name
-n : print nodename(hostname)
-v : print release
-m : print machine info
yeongju@vm-ubuntu20:~/sp2021-2/w06/q01$ ./systemSummary -s
This is a Linux machine.
yeongju@vm-ubuntu20:~/sp2021-2/w06/q01$ ./systemSummary -s -n
This is a Linux machine named vm-ubuntu20.
yeongju@vm-ubuntu20:~/sp2021-2/w06/q01$ ./systemSummary -s -n -v
This is a 5.11.0-37-generic Linux machine named vm-ubuntu20.
yeongju@vm-ubuntu20:~/sp2021-2/w06/q01$ ./systemSummary -s -n -v -m
This is a x86_64 5.11.0-37-generic Linux machine named vm-ubuntu20.
yeongju@vm-ubuntu20:~/sp2021-2/w06/q01$ ./systemSummary -n -m
This is a x86_64 machine named vm-ubuntu20.
```

[Q 2] Watchdog (shell 프로그래밍) [배점: 20]

hello.c 소스코드를 작성하시오. 소스코드 내의 main 함수를 아래와 같이 동작하도록 코딩하세요:

- 1) "Hello world begins" 메시지 출력
- 2) n 초 동안 sleep (n 은 10 에서 30 이내의 정수를 random 하게 선택. 즉, 매번 실행할 때 마다 다른 숫자가 선택되어야 함)
- 3) "Hello world ends" 메시지 출력

hello.c 를 컴파일 하여 hello.exe 실행파일을 생성 하시오.

[문제] hello.exe 프로그램이 종료하면 자동으로 재 시작 시키는 셸 스크립트 watchdog.sh 를 작성하시오. 셸 스크립트는 절대로 종료하지 않고, hello.exe 프로그램이 종료하면 hello.exe 를 재 시작 하고, 터미널에 "[shell] Restarting hello.exe!"라고 출력함. 즉, watchdog.sh 은 hello.exe 가 현재 실행중인 상태가 아니면 실행을 시키는 역할을 합니다.

[실행 순서]

- 1) 터미널에 date 명령 입력
- 2) watchdog.sh 스크립트를 약 2 분간 실행 (종료는 CTRL+C) // hello.exe 를 실행하지 않은 상태에서 watchdog.sh 를 실행하면 watchdog 은 hello.exe 를 실행시킴.
- 3) 터미널에 date 명령 입력 (1 번과 3 번에서 출력한 시간차가 2 분 또는 그 이상이 되어야 함)

터미널 화면을 캡처하고 본 문서에 첨부하시오. 소스코드(.c 및 .sh)는 첨부파일로 제출하시오.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w06/q02$ date  
2021. 10. 05. (화) 12:15:07 KST  
yeongju@vm-ubuntu20:~/sp2021-2/w06/q02$ ./watchdog.sh  
[shell] Restarting hello.exe!  
Hello world begins  
Hello world ends  
[shell] Restarting hello.exe!  
Hello world begins  
Hello world ends  
Hello world ends  
[shell] Restarting hello.exe!  
Hello world begins  
Hello world ends  
[shell] Restarting hello.exe!  
Hello world begins  
Hello world ends  
[shell] Restarting hello.exe!  
Hello world begins  
Hello world ends  
[shell] Restarting hello.exe!  
Hello world begins  
Hello world ends  
[shell] Restarting hello.exe!  
Hello world begins  
Hello world ends  
^C  
yeongju@vm-ubuntu20:~/sp2021-2/w06/q02$ date  
2021. 10. 05. (화) 12:17:31 KST
```

[Q 3] env [배점: 10]

아래와 같이 동작하는 findEnv.c 프로그램을 작성하세요

[사용 예] \$./findEnv PATH bin 라고 입력 시, PATH 환경변수에서 bin 이라는 문자열을 검색하고, 찾았다면 “PATH has bin!” 이라고 출력하고, 그렇지 않으면 “PATH does not have bin” 이라고 출력

[문제 1] \$./findEnv PATH bin 입력

[문제 1] \$./findEnv PATH hallym 입력

문제 1, 2와 같이 입력하고 터미널 출력 결과를 캡처하세요. 캡처 이미지를 본 문서에 첨부하고, 소스코드는 첨부파일로 제출하세요.

답변 1:

```
yeongju@vm-ubuntu20:~/sp2021-2/w06/q03$ ./findEnv PATH bin
PATH has bin!
```

답변 2:

```
yeongju@vm-ubuntu20:~/sp2021-2/w06/q03$ ./findEnv PATH hallym
PATH does not have hallym
```

[Q 4] 시간 측정, times [배점: 10]

아래와 같이 동작하는 measure_time.c 코드를 작성하세요.

[Task 1] $O(1)$ 시간이 소요되는 함수 func_01를 작성하고 실행시간을 측정 후 출력 하시오. 출력 메시지는 : func_01 takes xx seconds. (여기서 xx 대신 실제 측정된 시간을 초단위로 입력할 것)

[Task 2] $O(N)$ 시간이 소요되는 함수 func_ON를 작성하고 실행시간을 측정 후 출력 하시오. 출력 메시지는 : func_ON takes xx seconds. (여기서 xx 대신 실제 측정된 시간을 초단위로 입력할 것)

[Task 3] $O(N^2)$ 시간이 소요되는 함수 func_ON2를 작성하고 실행시간을 측정 후 출력 하시오. 출력 메시지는 : func_ON2 takes xx seconds. (여기서 xx 대신 실제 측정된 시간을 초단위로 입력할 것)

터미널 화면을 캡처하고 본 문서에 첨부하시오. 소스코드는 첨부파일로 제출하시오.

* 참고

- 위 프로그램은 동일한 값/배열/구조체 등을 입력으로 동작 합니다. 함수가 어떠한 동작을 하는지는 관계 없지만, 동일한 입력을 처리하도록 구현하세요.
- 위 프로그램의 실행 시간이 차이가 나도록, 입력 값을 충분히 크게/복잡하게 설정하세요 (예를 들어, 큰 사이즈의 배열을 입력으로 전달 ...).

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w06/q04$ ./measure_time
func_O1 takes 0.0000 seconds.
func_ON takes 0.0100 seconds.
func_ON2 takes 15.0100 seconds.
```

[Q 5] ncurses 따라하기 : 화면 중앙에 문자열 출력하기 [배점: 10]

아래의 예제 프로그램을 작성하고 실행해 보세요.


```

#include <ncurses.h>                /* ncurses.h includes stdio.h */
#include <string.h>

int main()
{
    char mesg[]="Just a string";      /* message to be appeared on the screen */
    int row,col;                     /* to store the number of rows and *
                                    * the number of columns of the screen */

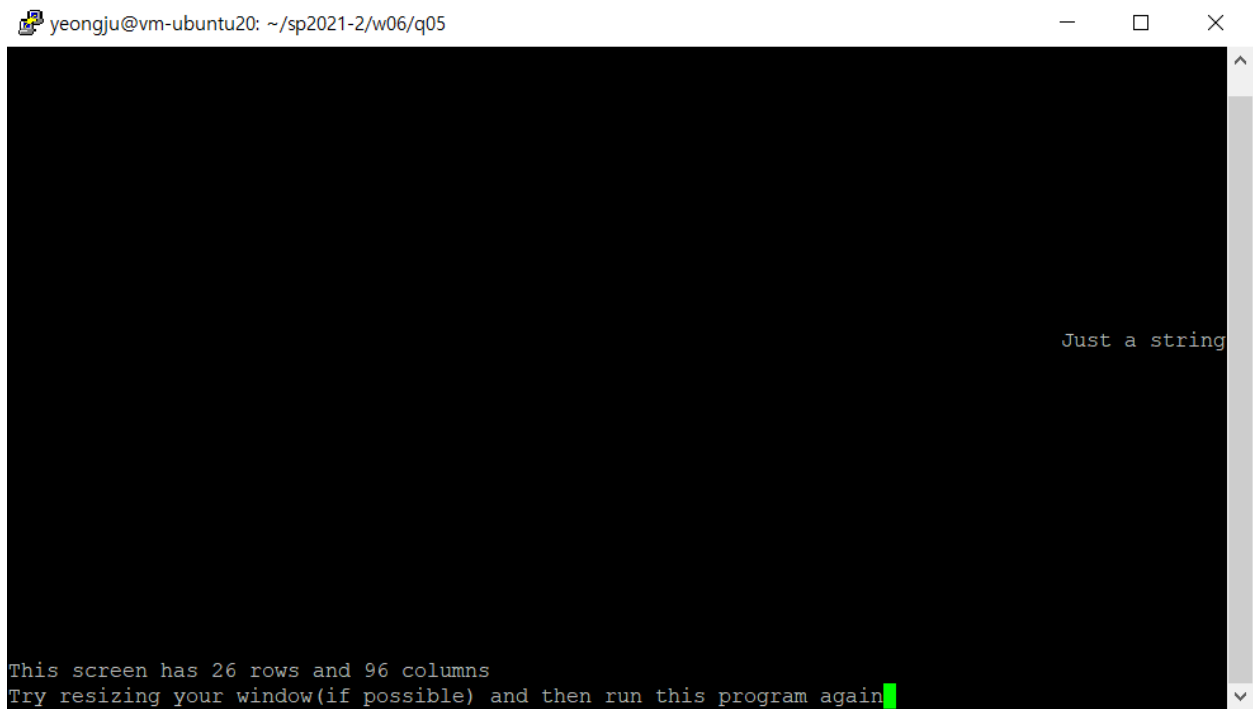
    initscr();                       /* start the curses mode */
    getmaxyx(stdscr,row,col);        /* get the number of rows and columns */
    mvprintw(row/2,(col-strlen(mesg))/2,"%s",mesg);
                                    /* print the message at the center of the screen */
    mvprintw(row-2,0,"This screen has %d rows and %d columns\n",row,col);
    printw("Try resizing your window(if possible) and then run this program again");
    refresh();
    getch();
    endwin();

    return 0;
}

```

‘mesg’ 문자열이 화면에서 무작위 위치에 나타나도록 하는 outputRandomPosition.c 프로그램을 작성하고 실행하세요. 터미널 출력 화면을 캡처해서 첨부하고, 소스코드도 첨부파일로 제출하세요.

답변:

A terminal window titled 'yeongju@vm-ubuntu20: ~/sp2021-2/w06/q05' with standard window controls. The terminal background is black with white text. The text 'Just a string' is displayed on the right side. At the bottom, a message reads: 'This screen has 26 rows and 96 columns' followed by 'Try resizing your window(if possible) and then run this program again' with a green cursor at the end.

```
yeongju@vm-ubuntu20: ~/sp2021-2/w06/q05
```

Just a string

This screen has 26 rows and 96 columns
Try resizing your window(if possible) and then run this program again

[Q 6] ncurses 따라하기 : 입력 받기 (inputExample.c) [배점: 10]

아래의 프로그램을 입력하고 실행하세요.

```
#include <ncurses.h>                /* ncurses.h includes stdio.h */
#include <string.h>

int main()
{
    char mesg[]="Enter a string: ";    /* message to be appeared on the screen */
    char str[80];
    int row,col;                       /* to store the number of rows and *
                                       * the number of colums of the screen */

    initscr();                         /* start the curses mode */
    getmaxyx(stdscr,row,col);          /* get the number of rows and columns */
    mvprintw(row/2,(col-strlen(mesg))/2,"%s",mesg);
                                       /* print the message at the center of the screen */

    getstr(str);
    mvprintw(LINES - 2, 0, "You Entered: %s", str);
    getch();
    endwin();

    return 0;
}
```

터미널 출력 화면을 캡처해서 첨부하세요. 소스코드는 첨부파일로 제출하세요.

답변:



A terminal window titled 'yeongju@vm-ubuntu20: ~/sp2021-2/w06/q06' with standard window controls. The terminal background is black. It displays the prompt 'Enter a string: ' followed by the user input 'hello'. Below this, it shows 'You Entered: hello' with a green cursor at the end.

```
yeongju@vm-ubuntu20: ~/sp2021-2/w06/q06
Enter a string: hello
You Entered: hello
```

[Q 7] ncurses 따라하기 : 컬러 속성 (attributeEx.c) [배점: 10]

아래의 프로그램을 입력하고 실행하세요.

```

#include <ncurses.h>

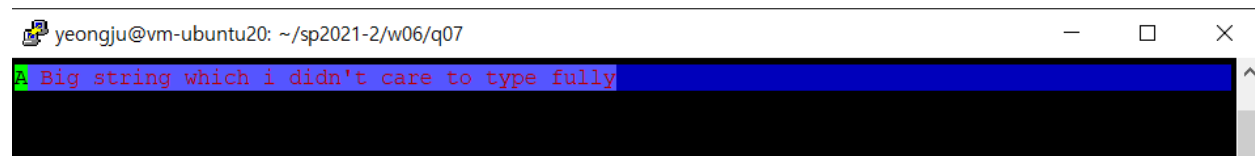
int main(int argc, char *argv[])
{
    initscr();                /* Start curses mode      */
    start_color();            /* Start color functionality */

    init_pair(1, COLOR_CYAN, COLOR_BLACK);
    printw("A Big string which i didn't care to type fully ");
    mvchgat(0, 0, -1, A_BLINK, 1, NULL);
    /*
     * First two parameters specify the position at which to start
     * Third parameter number of characters to update. -1 means till
     * end of line
     * Forth parameter is the normal attribute you wanted to give
     * to the character
     * Fifth is the color index. It is the index given during init_pair()
     * use 0 if you didn't want color
     * Sixth one is always NULL
     */
    refresh();
    getch();
    endwin();                 /* End curses mode      */
    return 0;
}

```

색 조합을 (RED, BLUE)로 변경하고 프로그램을 실행하세요. 터미널 출력 화면을 캡처해서 첨부하세요. 소스코드는 첨부파일로 제출하세요.

답변:



[Q 8] ncurses 따라하기 : 윈도우 (borderEx.c) [배점: 10]

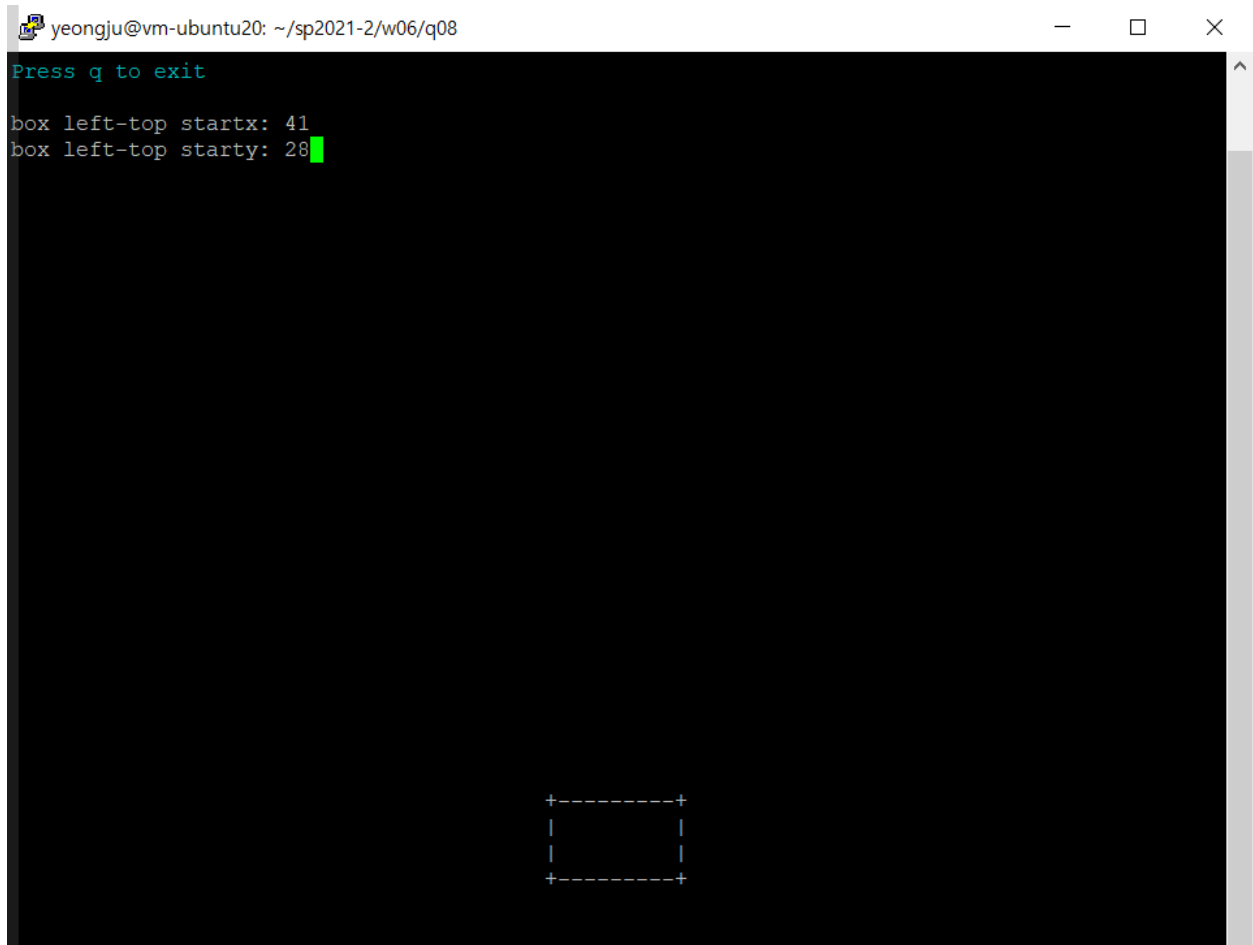
예제 파일 borderEx.c 프로그램을 실행하세요. 프로그램을 다음과 같이 변경하세요.

- ‘i’ (=information) 키가 눌리면 화면 아래에, BOX의 좌측 상단 좌표 (starty, startx)를 출력하세요.

터미널 출력 화면을 캡처해서 첨부하세요. 소스코드는 첨부파일로 제출하세요.

(borderEx.c 다운로드 링크 : <https://github.com/overegoz/sysprog/blob/master/w06-lab/q8-ncurses/>)

답변:



```
yeongju@vm-ubuntu20: ~/sp2021-2/w06/q08
Press q to exit
box left-top startx: 41
box left-top starty: 28
+-----+
|       |
|       |
+-----+
```

끝! 수고하셨습니다 ☺