System Programming
(ELEC462)

Lab #6

Dukyun Nam

HPC Lab@KNU

Lab #6-1: Writing `write0`

- Write `write0`
  - Source code for submission: `write0.c`
  - Make sure that your code must work properly

Lab #6-2: Use the `tty` driver

- Write `echostate` and `setecho`
  - Source code for submission: `echostate.c` and `setecho.c`
  - Make sure that your code must work properly

Lab #6-3: Programming the terminal driver

- Write `showstty`
  - Source code for submission: `showstty.c`
  - Make sure that your code must work properly

Lab #6-4: Use the size of the terminal screen

- Write `more03`
  - Modify the version of `more01` so that is uses the size of the terminal screen instead of the fixed value of 24
  - Source code for submission: `more03.c`
  - Make sure that your code must work properly

Lab #6-4: more01.c

```c
/* more01.c  - version 0.1 of more
 *      read and print 24 lines then pause for a few special commands
 */

#include        <stdio.h>
#include        <stdlib.h>

#define PAGELEN 24
#define LINELEN 512

void do_more(FILE *);
int  see_more();

int main( int ac , char *av[] )
{
        FILE    *fp;

        if ( ac == 1 )
                do_more( stdin );
        else
                while ( --ac )
                        if ( (fp = fopen( *++av , "r" )) != NULL )
                        {
                                do_more( fp ) ;
                                fclose( fp );
                        }
                        else
                                exit(1);
        return 0;
}
```

Lab #6-4: `more01.c` (cont.)

```c
void do_more( FILE *fp )
/*
 *  read PAGELEN lines, then call see_more() for further instructions
 */
{
        char    line[LINELEN];
        int     num_of_lines = 0;
        int     see_more(), reply;

        while ( fgets( line, LINELEN, fp ) ){           /* more input   */
                if ( num_of_lines == PAGELEN ) {        /* full screen? */
                        reply = see_more();             /* y: ask user  */
                        if ( reply == 0 )               /*    n: done   */
                                break;
                        num_of_lines -= reply;          /* reset count  */
                }
                if ( fputs( line, stdout )  == EOF )    /* show line    */
                        exit(1);                        /* or die       */
                num_of_lines++;                         /* count it     */
        }
}
```

```c
int see_more()
/*
 *      print message, wait for response, return # of lines to advance
 *      q means no, space means yes, CR means one line
 */
{
        int     c;

        printf("\033[7m more? \033[m");         /* reverse on a vt100   */
        while( (c=getchar()) != EOF )                   /* get response */
        {
                if ( c == 'q' )                 /* q -> N               */
                        return 0;
                if ( c == ' ' )                 /* ' ' => next page     */
                        return PAGELEN;         /* how many to show     */
                if ( c == '\n' )                /* Enter key => 1 line  */
                        return 1;
        }
        return 0;
}
```

Lab #6: Submission

- Deadline: Tomorrow 11:59pm
  - Create a directory name (`lab6`) to another using a series of the following commands:
    - `mkdir lab6_s<Your_Student_ID>`
      - Assume your ID is 2022000000.
  - Zip your folder:
    - `zip -r lab6_s2022000000.zip lab6_s2022000000`
- Upload the zipped directory (`lab6_s2022000000.zip`) into LMS