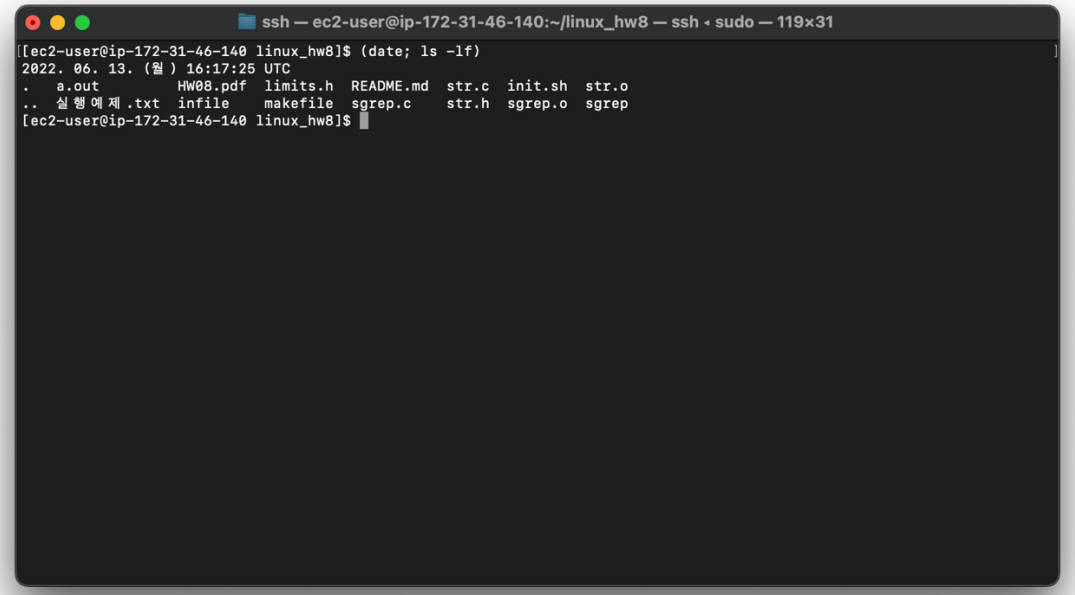
리눅스시스템및실습 과제 #8 (HW_8)

214823 컴퓨터정보통신공학과 박종현

1. linux_hw8(linux_hw7) 디렉터리에서 (date; ls -lf)를 실행한 결과



2. sgrep.c 프로그램 소스 코드와 Makefile 내용

sgrep.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h> /* for skeleton code */
#include <unistd.h> /* for getopt */
#include "str.h"
#define FALSE 0
#define TRUE 1
/*
* Fill out your own functions here (If you need)
*/
/*-------------------------------------------------------------------------*/
/* PrintUsage()
print out the usage of the Simple Grep Program */
/*-------------------------------------------------------------------------*/
void PrintUsage(const char* argv0)
{
```

```
  const static char *fmt =
    "Simple match (match) Usage:\n"
    "%s pattern \n";
  printf(fmt, argv0);
}

/*--------------------------------------------------------------------*/
/* SearchPattern()
Your task:
1. Do argument and input string validation - String or file argument length is no
more than 1023 - If you encounter an input argument that's too long, print out
"Error: pattern is too long"
2. Read the each line from input file(infile) - If you encounter a line larger than
1023 bytes, print out "Error: input line is too long" - Error message should be
printed out to standard error (stderr)
3. Check & print out the line contains a given string (search-string)
Tips: - fgets() is an useful function to read characters from file. Note
that the fget() reads until newline or the end-of-file is reached. -
fprintf(sderr, ...) should be useful for printing out error
message to standard error
NOTE: If there is any problem, return FALSE; if not, return TRUE */
/*--------------------------------------------------------------------*/
int SearchPattern(const char *pattern)
{
  char buf[MAX_STR_LEN + 2];
  FILE *fp;
  if( (fp = fopen("infile", "r")) == NULL) {
    fprintf(stderr, "Error: file open error\n");
    return(EXIT_FAILURE);
  }
  /*
  * TODO: check if pattern is too long and there exists
  */

  int lines = 0;
  int matched = 0;
  while (fgets(buf, MAX_STR_LEN + 2, fp) != NULL) {
    char *lowered = StrToLower(buf);
    lines++;
    for (int i = 0; i < StrGetLength(lowered) - StrGetLength(pattern); i++) {
      int flag = 0;
      for (int j = 0; j < StrGetLength(pattern); j++) {
        if (lowered[i+j] != pattern[j]) {
          flag = 0;
```

```
          break;
        } else {
          flag = 1;
        }
      }
      if (flag) {
        matched = 1;
        printf("(ln %d, pos %d) %s", lines, i+1, buf);
        if (buf[StrGetLength(buf)-1] != '\n') printf("\n");
      }
    }
  }
  if (!matched) {
    fprintf(stderr, "No pattern\n");
    return FALSE;
  }
  return TRUE;

}

/*----------------------------------------------------------------------*/
int main(const int argc, const char *argv[])
{
  /* Do argument check and parsing */
  if (argc <= 1) {
    fprintf(stderr, "Error: argument parsing error\n");
    PrintUsage(argv[0]);
    return EXIT_FAILURE;
  }
  if (StrGetLength(argv[1]) > MAX_STR_LEN) {
    fprintf(stderr, "Too long\n");
    return EXIT_FAILURE;
  }
  return SearchPattern(StrToLower(argv[1])) ? EXIT_SUCCESS : EXIT_FAILURE;
}
```

* 과제에서 제시한 1번 항목의 두번째, <문자열과 관련한 함수를 포함하는 프로그램>과 sgrep.c에서 사용하는 함수의 작동 사양을 만족시키기 위해 str.c의 StrToLower함수를 아래와 같이 수정함

```
char *StrToLower(char *str)
{
  char *str_clone;
  str_clone = (char *)malloc(1024);
```

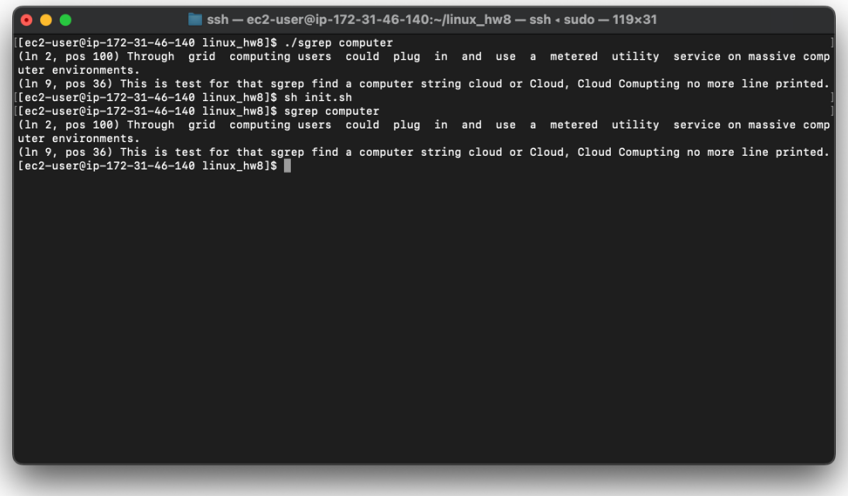
```
  int count = 0;
  /* TODO: fill this function */
  /* Part 1 */
  for (int i = 0; i < MAX_STR_LEN; i++) {
    if (str[i] >= 65 && str[i] <= 90) {
      str_clone[i] = str[i] + 32;
    } else {
      str_clone[i] = str[i];
    }
  }
  return str_clone;
}
```

makefile

```
cc = gcc

sgrep: sgrep.o str.o
  $(cc) -g -o sgrep sgrep.o str.o
sgrep.o: sgrep.c str.c str.h limits.h
  $(cc) -c sgrep.c
str.o: str.c str.h limits.h
  $(cc) -c str.c
```

3. 고정된 파일(infile)과 입력받은 문자열에 대해 sgrep을 실행한 결과를 캡쳐한 파일



* 현재 경로 위의 파일을 실행하려면 ./ 문자열을 앞에 붙여서 실행해야함. 따라서 과제에서 제시한 목표를 달성하기 위해 init.sh 셸 스크립트 파일을 별도로 작성하고, 과제에서 제시한 명령을 실행하기 전에 우선 실행함

init.sh

```
# Temporary path setter
CURRENT=`pwd`
export PATH="$CURRENT:$PATH"
```