컴퓨터학부 20152385 송민구

1. 소스 코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/time.h>

#include "ssu\_runtime.h"

#define TABLE\_SIZE 128

#define BUFFER\_SIZE 1024

int main(void)

{

struct timeval begin\_t, end\_t;

static struct {

long offset;

int length;

} table[TABLE\_SIZE];

char buf[BUFFER\_SIZE];

long offset;

int entry;

int i;

int length;

int fd;

gettimeofday(&begin\_t, NULL);

if (argc < 2) { // 프로그램 실행 시 파일명이 입력 되지 않았다면

fprintf(stderr, "usage : %s <file>\n", argv[0]); // 오류 출력 후

exit(1); // 종료

}

if ((fd = open(argv[1], O\_RDONLY)) < 0) { // 읽기 권한으로 open()

fprintf(stderr, "open error %s\n", argv[1]); // 실패 시 오류 출력 후

exit(1); // 종료

}

entry = 0;

offset = 0;

while ((length = read(fd, buf, BUFFER\_SIZE)) > 0) { // 버퍼의 사이즈만 큼 buf에 입력받고 길이를 length에 저장

for (i = 0; i < length; i++) { // 받은 길이만큼

table[entry].length++; // 한 줄의 length 증가

offset++; // offset증가

if (buf[i] == '\n') { // 개행을 만나면

table[++entry].offset = offset; // 다음 줄로 이동 후 offset저장 (그 줄의 첫 글자)

}

}

}

#ifdef DEBUG

for (i = 0; i < entry; i++) { // 디버깅을 위한 출력

printf("%d : %ld, %d\n", i + 1, table[i].offset, table[i].length);

}

#endif

while (1) {

printf("Enter line number : ");

scanf("%d", &length);

if (--length < 0) { // 입력 값이 0보다 작다면

break; // 반복문 탈출

}

lseek(fd, table[length].offset, 0); // 입력된 숫자만큼 줄을 이동

if (read(fd, buf, table[length].length) <= 0) { // 그 줄의 길이가 0보다 작다면

continue; // 다시 입력

}

buf[table[length].length] = '\0'; // 맨 마지막 칸에 NULL삽입 후

printf("%s", buf); // 문자열 출력

}

close(fd); // 파일을 닫음

gettimeofday(&end\_t, NULL);

ssu\_runtime(&begin\_t, &end\_t);

exit(0);

}

1. 실행 결과

