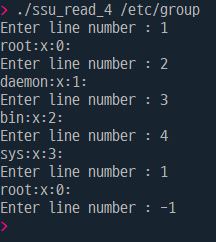
컴퓨터학부 20162448 김병준

1. 결과



1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <fcntl.h>  #define TABLE\_SIZE 128 // 구조체를 저장할 테이블 크기  #define BUFFER\_SIZE 1024 // 버퍼 크기  int main(int argc, char \*argv[])  {  static struct {  long offset;  int length;  } table[TABLE\_SIZE]; // 테이블 데이터 구조체  char buf[BUFFER\_SIZE];  long offset;  int entry;  int i;  int length;  int fd;  if(argc < 2) { // 인자 개수가 부족할 경우  fprintf(stderr, "usage : %s <file>\n", argv[0]);  exit(1);  }  if((fd = open(argv[1], O\_RDONLY)) < 0) { // 인자로 주어진 파일을 읽기 전용으로 열기  fprintf(stderr, "open error for %s\n", argv[1]);  exit(1);  }  entry = 0;  offset = 0;  while((length = read(fd, buf, BUFFER\_SIZE)) > 0) { // 파일에 대하여 1024바이트만큼 버퍼에 읽어들임  for(i = 0; i < length; i++) {  table[entry].length++; // 테이블 크기 증가  offset++; // 오프셋 증가  if(buf[i] == '\n') // 개행문자일 경우  table[++entry].offset = offset; // 요소 개수 증가 및 현재 요소의 오프셋 위치 지정  }  }  #ifdef DEBUG  for(i = 0; i < entry; i++)  printf("%d : %ld, %d\n", i + 1, table[i].offset, table.length);  #endif  while(1) {  printf("Enter line number : ");  scanf("%d", &length);  if(--length < 0) // 입력이 0보다 작을 경우  break;  lseek(fd, table[length].offset, 0); // 오프셋을 입력한 --라인\*데이터 만큼 이동  if(read(fd, buf, table[length].length) <= 0) // 파일에서 해당 위치의 라인을 읽어들임  continue;  buf[table[length].length] = '\0'; // 끝에 널값으로 문자열의 끝을 지정  printf("%s", buf);  }    close(fd); // 디스크럽터 연결 종료  exit(0);  } |