16강

C_PROGRAMMING

file 입출력

❖ File descriptor

- File에 관한 내용들을 system입장에 user에게 사용하기 편하게 부여한 ID

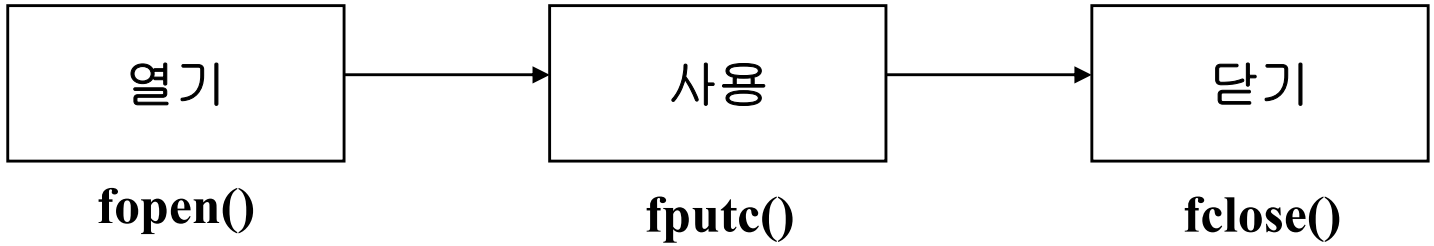
File descriptor	file	설명
0	stdin	표준 입력
1	stdout	표준 출력
2	stderr	표준 에러

FILE 정의

```
typedef struct   {
        short               level;     /* fill/empty level of buffer */
        unsigned            flags;    /* File status flags    */
        char                fd;        /* File descriptor      */
        unsigned char       hold      /* Ungetc char if no buffer */
        short               bsize;   /* Buffer size          */
        unsigned char       *buffer;/* Data transfer buffer */
        unsigned char       *curp;  /* Current active pointer */
        unsigned            istemp;/* Temporary file indicator */
        short               token; /* Used for validity checking */
}       FILE;                              /* This is the FILE object */
```

file 입출력

❖ file을 사용하기 위한 절차



❖ file 열기

- 형식
  - FILE *fopen(const char **filename*, const char **mode*);
  - filename: 읽어들일 file명
  - mode : 다음페이지 참조
  - 반환값 : stream으로 값을 반환함

file 접근 방식의 종류

종류	지정한 file명의 file이 현재 존재하는 경우	지정한 file명의 file이 현재 존재하지 않는 경우
“r”	file을 읽기(read) 전용으로 개방한다.	에러가 발생했다는 의미로 NULL이 반환된다.
“w”	현재의 내용을 삭제하고, 쓰기 전용으로 개방한다.	새로운 file을 생성하고, 쓰기 전용으로 개방한다.
“a”	file에 겹쳐 쓰지 않고, 기존 file의 끝에 쓰기 전용으로 개방한다.	새로운 file을 생성하고, 쓰기 전용으로 개방한다.
“r+”	file을 읽기와 쓰기용으로 개방한다.	에러가 발생했다는 의미로 NULL이 반환된다.
“w+”	현재의 내용을 삭제하고 읽기 쓰기로 개방한다.	새로운 file을 생성하고, 읽기 쓰기용으로 개방한다.
“a+”	file에 겹쳐쓰지 않고 기존 file의 끝에서 읽기와 쓰기용으로 개방한다.	새로운 file을 생성하고, 읽기 쓰기용으로 개방한다.

사용예

- fp = fopen("test.txt", "rb");
- fp = fopen("test.txt", "w");
- fp = fopen("test.txt", "a+");
- if ((fp=fopen("test.txt", "rb")) == NULL) {
  printf("File open error.₩n");
  }

file 입력

- file 사용
  - 한 문자 입력
    - int fgetc(FILE *fp);
  - 서식화 입력
    - int fscanf(FILE *fp, const cahr *format, ...);
  - 블록 입력
    - size_t fread(void *ptr, size_t size, size_t n, FILE *fp);
  - 문자열 입력
    - char *fgets(char *s, int n, FILE *fp);

file 출력

- file 사용
  - 한 문자 출력
    - int fputc(int c, FILE *fp)
  - 서식화 출력
    - int fprintf(FILE *fp, const char *format, ...);
  - 블록 출력
    - size_t fwrite(const void *ptr, size_t size, size_t n, FILE *fp);
  - 문자열 출력
    - int fputs(const char *s, FILE *fp);

- file 닫기
  - int fclose(FILE *fp);

사용예

```
#include <stdio.h>
void main()
{
   FILE *test;
   test=fopen("c:₩₩test.txt", "w");   //test.txtfile 생성, 쓰기 설정
   fprintf(test, "%s", "hi");         //test.txt에 hi입력
   fclose(test);                      //종료
}
```

Quiz

❖ Test.txt 파일에 다음 내용을 저장 하시오

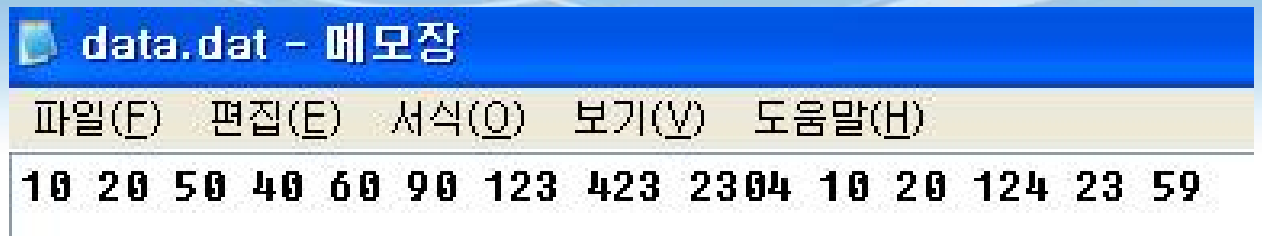
이름 : 홍길동
나이 : 16
주소 : 산골짜기

fprintf

```
#include <stdio.h>

void main(void) {
        FILE *fp;
        char str[80];
        int line = 0;
        if ((fp = fopen("C:₩₩test.txt", "w")) == NULL) {
                printf("File open error ... ₩n");
                return;
        }
       while(1) {
                gets(str);
                if (str[0] == '₩0')
                        break;
                line++;
                fprintf(fp,"%3d : %s₩n",line,str);
        }
        fclose(fp);
}
```

fscanf




```
#include <stdio.h>
main() {
        FILE *fp;
        int i, no = 0, sum = 0;
        if ((fp = fopen("C:₩₩data.dat", "r")) == NULL) {
                printf("Error: Cannot open data.dat₩n");
                return 1;
        }
        while (fscanf(fp,"%d",&i) != EOF) {
                sum += i;
                no++;
        }
        if (no == 0) printf("No data.₩n");
        else {
                printf("Total %d numbers.₩n",no);
                printf("Sum = %ld₩n",sum);
                printf("Average = %.2f₩n",(float)sum/(float)no);
        }
        fclose(fp);
}
```

main 함수 인자 값

❖ main 함수에서 사용되는 인자 값 argc와 argv[]는 프로그램 실행 시 특정 값을 입력할 때 사용된다.

❖ 예

```
#include <stdio.h>
int main(int argc, char *argv[])
{
        int i;
        printf("argc: %d₩n", argc);
        for(i=0;i<argc;i++)
                printf("argv[%d]: %s₩n", i, argv[i]);
        return 0;
}
```

```
myfile.exe -c filename1 filename2
argc: 4
argv[0]: myfile.exe
argv[1]: -c
argv[2]: filename1
argv[3]: filename2
```