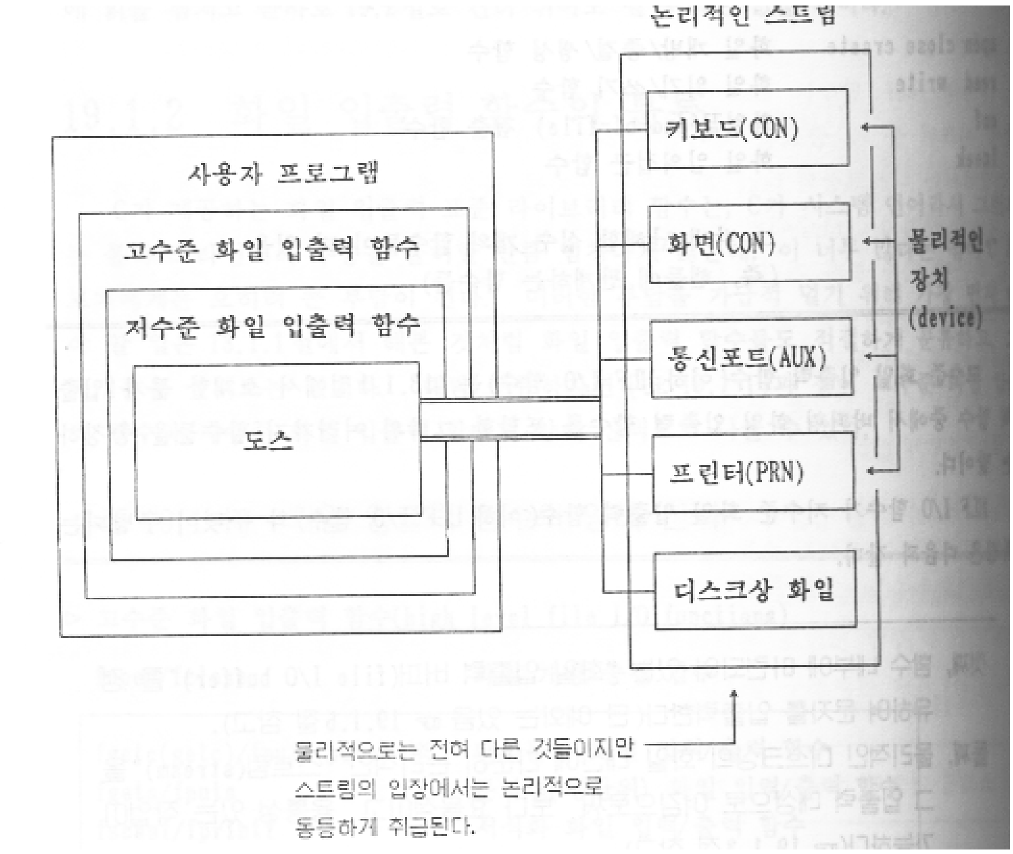
9. File input and output

9.1 Stream and file I/O buffer

* stream : 물리적인 디스크상의 파일과 도스 장치들을 하나의 통일된 방식으로 다루기 위한 가상의 장치를 말함.



물리적으로는 전혀 다른 것들이지만
스트림의 입장에서는 논리적으로
동등하게 취급된다.

* stream 의 사용방법

- stream variable declaration	: FILE *fp ;
- open stream	: fp=fopen(“test.dat”,”r”) ;
- I/O operation	: fgets(fp,s,80) ;
- close stream	: fclose(fp) ;

* file I/O buffer

- 일반적으로 I/O operation 은 시간이 많이 소요됨
- 일정량을 담아 두었다가 일시에 입/출력을 행하는 것이 유리함

  (buffered I/O operation)

* for DOS applications

```
config.sys
        files  = 20        → number of maximum open files
        buffer  = 512      → size of file I/O buffer
```

9.2 FILE structure and file pointer

```
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 */
```

- 파일구조체에 대한 상세한 내용을 알 필요는 없음; 사용방법만 고려

```
FILE *fp ;
fp = fopen(“test.dat”,”r”) ;          → 파일 구조체를 위한 메모리 할당, 파일 열기
fclose(fp) ;                          → 메모리 반납, 파일 닫기
```

9.3 File access mode

* file access mode

- r : read only
- w : write only (new stream)
- a : append
- r+ : read and update
- w+ : write and update
- a+ : append and update

* Character I/O mode

- t : text mode : CR/LF 변환 수행
- b : binary mode : 아무런 변환을 거치지 안흠

Example 1) file copy example

```
#include <stdio.h>
void main(int argc, char *argv[])
{
        int c ;
        FILE *src, *dest ;

        src = fopen(argv[1],"rb") ;
        dest = fopen(argv[2],"wb") ;
        while ( (c=fgetc(src)) != EOF)
                fputc(c,dest) ;
}
```

```
c:\ TC>filecopy filename1 filename2
```

9.4 Basic file I/O functions

- fgetc, fputc : 단일 문자 입출력 함수

```
int fgetc(FILE *stream) ;
int fputc(int c,FILE *stream) ;
```

- fgets, fputs : 행단위 문자열 입출력 함수

```
char *fgets(char *s, int n, FILE *stream) ;
int fputs(char *s, FILE *stream) ;
```

- fprintf, fscanf : 서식화 입출력 함수

```
int fscanf(FILE *stream, const char *format, ...) ;
inf fprintf(FILE *stream, const char *format, ...) ;
```

Example 2) file printing example

```
#include <stdio.h>
#define MAXLEN 80

void main()
{
        FILE *fp ;
        char *str[MAXLEN +1 ] ;
        fp = open(“test.dat”,”r”) ;
        while( fgets(str,MAXLEN,fp) != NULL)     puts(str) ;
        return ;
}
```