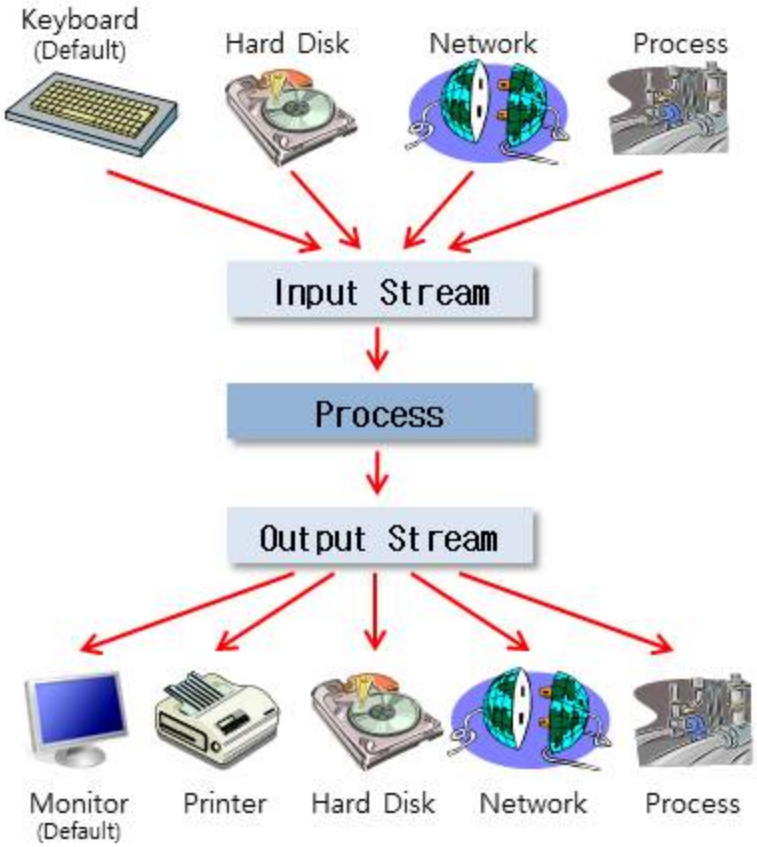
C언어 강의자료

문정욱

프로그래밍 연습 7
파일 입출력

스트림

- Stream
  - 데이터 입출력 통로

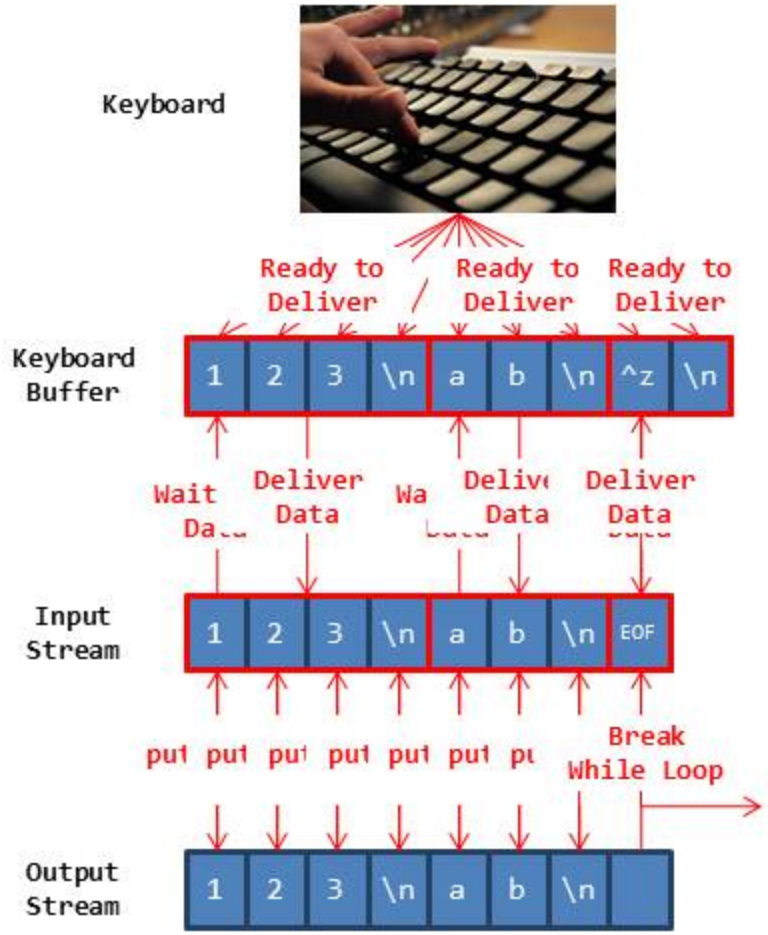



```c
#include <stdio.h>

int main(void)
{
    int ch;

    while( (ch=getchar())!=EOF ) {
        putchar(ch);
    }
    return 0;
}
```

```c
#include <stdio.h>

int main(void)
{
    int ch;

    while( (ch=fgetc(stdin))!=EOF ) {
        fputc(ch,stdout);
    }
    return 0;
}
```

스트림과 키보드

- 키보드를 통한 정보 입력



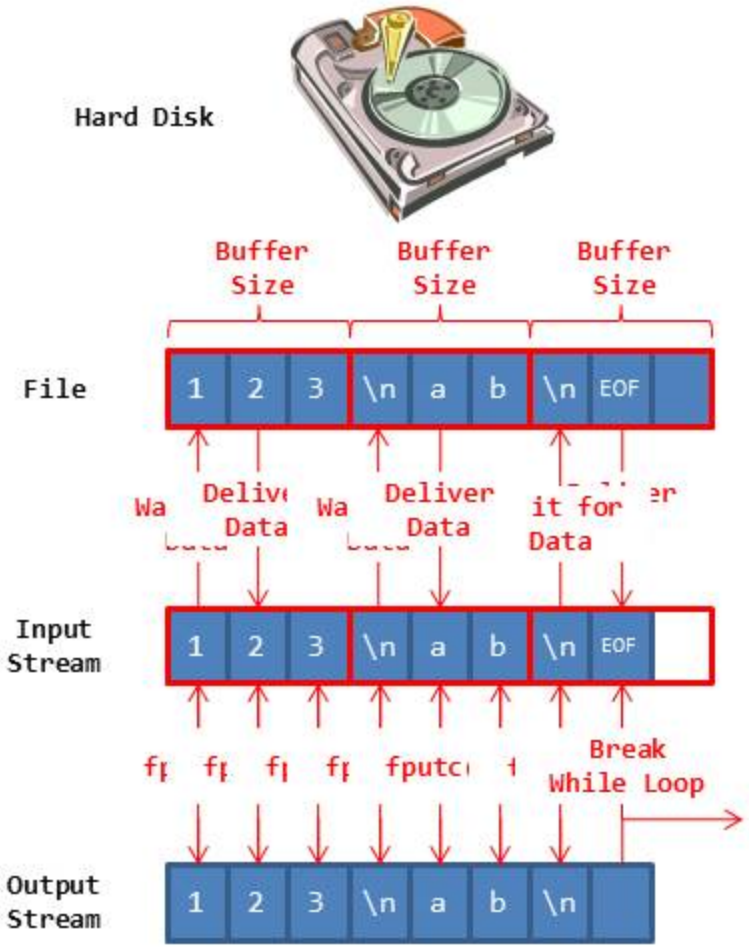

```
#include <stdio.h>

int main(void)
{
    int ch;

    while( (ch=getchar())!=EOF ) {
        putchar(ch);
    }
    return 0;
}
```

입출력 결과

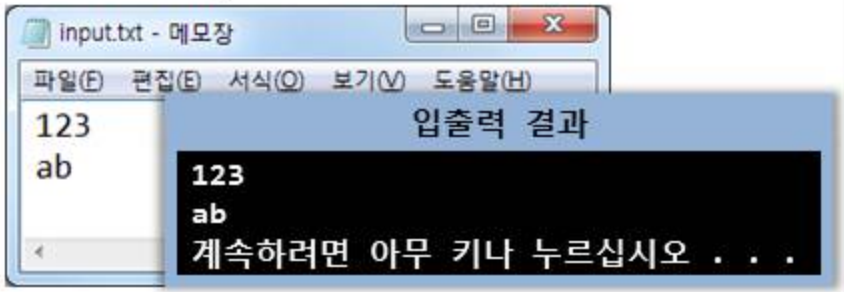
```
123
123
ab
ab
^Z
계속하려면 아무 키나 누르십시오 . . .
```

스트림과 파일

- 파일을 통한 정보 입력

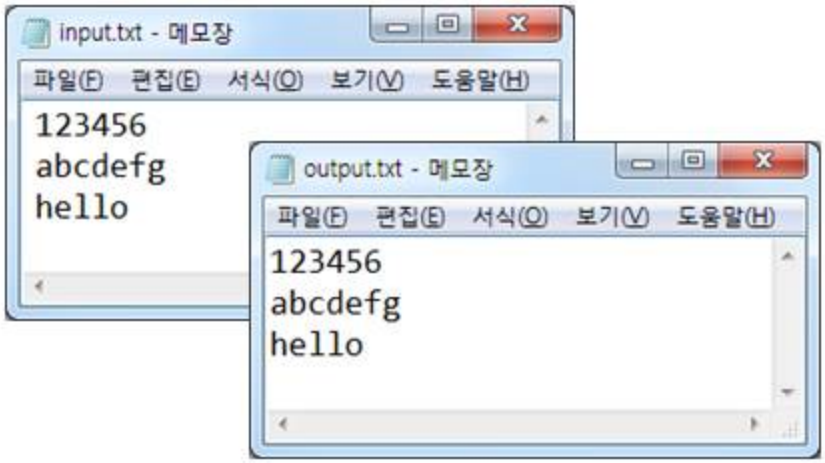



```
#include <stdio.h>

int main(void)
{
    FILE* fp;
    int ch;

    fp=fopen("input.txt","rt");

    while( (ch=fgetc(fp))!=EOF ) {
        fputc(ch,stdout);
    }

    fclose(fp);
    return 0;
}
```



파일 입출력

- 파일 복사
  - 문자의 입출력




```c
#include <stdio.h>

int main(void)
{
    FILE* infp;
    FILE* outfp;
    int ch;

    infp=fopen("input.txt","rt");
    outfp=fopen("output.txt","wt");

    while( (ch=fgetc(infp))!=EOF ) {
        fputc(ch,outfp);
    }

    fclose(outfp);
    fclose(infp);
    return 0;
}
```

FILE 구조체

- FILE 구조체
  - 요구조건
    - stdio.h
  - 정보
    - 열린 파일의 상태 정보 보유
  - 내부 구조
    - 컴파일러마다 다름
    - FILE 구조체 내부 변수를 참조하는 것은 바람직하지 못하다.

```
struct _iobuf {
    char *_ptr;
    int   _cnt;
    char *_base;
    int   _flag;
    int   _file;
    int   _charbuf;
    int   _bufsiz;
    char *_tmpfname;
};
typedef struct _iobuf FILE;
```

fopen() 함수의 사용 from MSDN 2005

Open mode

modes	descriptions
"r"	Opens for reading. If the file does not exist or cannot be found, the **fopen** call fails.
"w"	Opens an empty file for writing. If the given file exists, its contents are destroyed.
"a"	Opens for writing at the end of the file (appending) without removing the EOF marker before writing new data to the file; creates the file first if it doesn't exist.
"r+"	Opens for both reading and writing. (The file must exist.)
"w+"	Opens an empty file for both reading and writing. If the given file exists, its contents are destroyed.
"a+"	Opens for reading and appending; the appending operation includes the removal of the EOF marker before new data is written to the file and the EOF marker is restored after writing is complete; creates the file first if it doesn't exist.

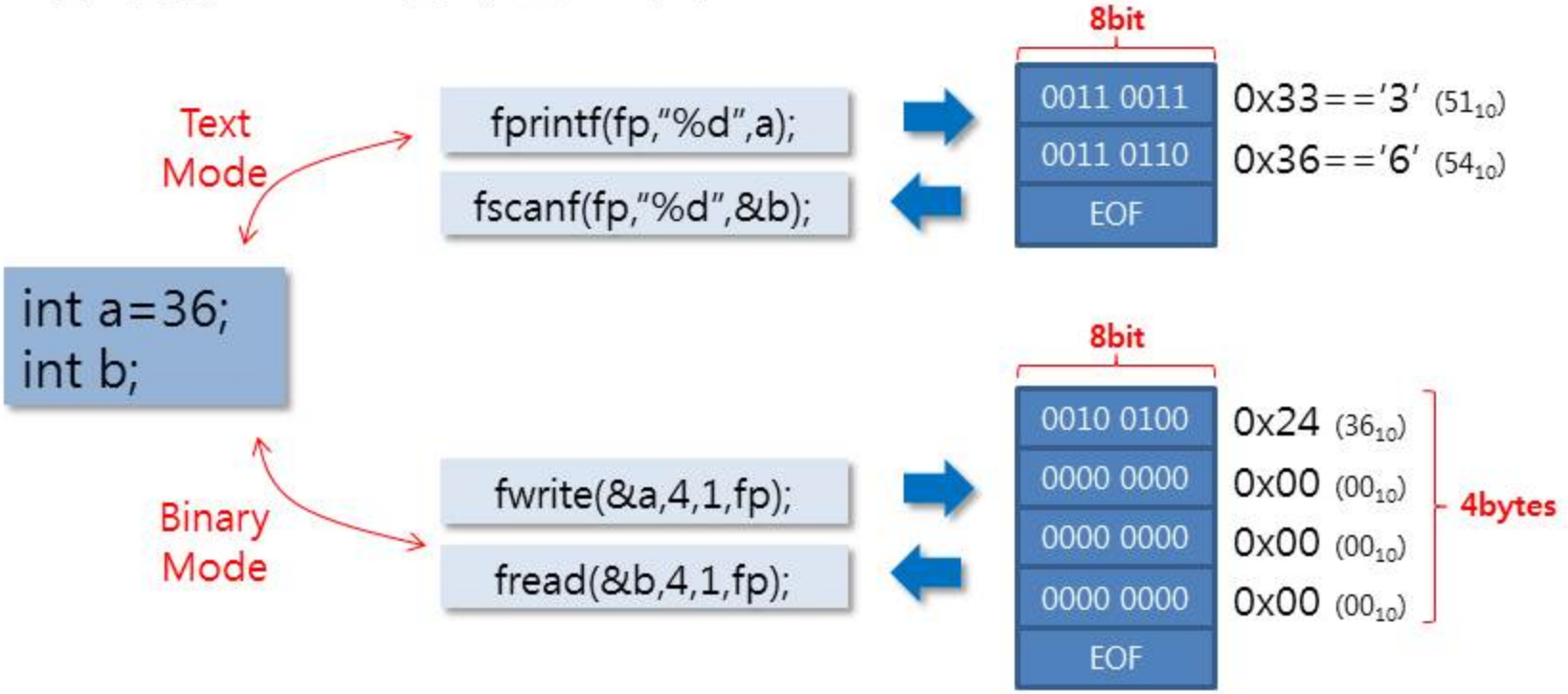
fopen() 함수의 사용 from MSDN 2005

Types

types	descriptions
t	Open in text (translated) mode. In this mode, CTRL+Z is interpreted as an end-of-file character on input. In files opened for reading/writing with **"a+"**, **fopen** checks for a CTRL+Z at the end of the file and removes it, if possible. This is done because using **fseek** and **ftell** to move within a file that ends with a CTRL+Z, may cause **fseek** to behave improperly near the end of the file. Also, in text mode, carriage return–linefeed combinations are translated into single linefeeds on input, and linefeed characters are translated to carriage return–linefeed combinations on output.
b	Open in binary (untranslated) mode; translations involving carriage-return and linefeed characters are suppressed.

텍스트 모드와 바이너리 모드

- Text mode와 Binary mode의 저장 방식 비교
  - Text mode: 문자 코드로 정보를 저장한다. 그러므로, 텍스트 에디터(text editor)로 내용 확인 및 편집이 가능하다.
  - Binary mode: 이진 정보 그대로 저장한다. 내용확인과 편집을 위해서는 전용 프로그램이 필요하다.



Character Constant

- ASCII Code
  - American Standard Code for Information Interchange Code

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

fclose() 함수의 사용 from MSDN 2005

- int fclose(FILE *stream);
  - Closes a stream
  - Parameters
    - stream: Pointer to FILE structure.
  - Return Value
    - fclose() returns 0 if the stream is successfully closed. This function returns EOF to indicate an error.
  - Remarks
    - The fclose() function closes stream.
  - Requirements
    - <stdio.h>

파일 입출력 예외처리

- 예외처리
  - fopen에 의해서 리턴 받은 스트림이 NULL일 경우 파일 열기에 실패했음을 뜻하므로
  - 계획된 작업을 중지하고 이에 대한 오류 메시지 출력과 적절한 작업처리가 필요하다.

입출력 결과

```
123456
abcdefg
hello
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
File cannot opened.
계속하려면 아무 키나 누르십시오 . . .
```

```c
#include <stdio.h>

int main(void)
{
    FILE* fp;
    int ch;

    fp=fopen("input.txt","rt");
    if(fp==NULL) {
        printf("File cannot opened.\n");
        return -1;
    }

    while( (ch=fgetc(fp))!=EOF ) {
        fputc(ch,stdout);
    }

    fclose(fp);
    return 0;
}
```

파일을 통한 문자 단위 입출력

File Stream

```c
#include <stdio.h>

int main(void)
{
    FILE* infp;
    FILE* outfp;
    int ch;

    infp=fopen("input.txt","rt");
    outfp=fopen("output.txt","wt");

    while( (ch=fgetc(infp))!=EOF ) {
        fputc(ch,outfp);
    }

    fclose(outfp);
    fclose(infp);
    return 0;
}
```

Standard Stream

```c
#include <stdio.h>

int main(void)
{
    int ch;

    while( (ch=getchar())!=EOF ) {
        putchar(ch);
    }

    return 0;
}
```

파일을 통한 문자열 단위 입출력

File Stream

```c
#include <stdio.h>

int main(void)
{
    FILE* infp;
    FILE* outfp;
    char str[512];
    int max_size=sizeof(str);

    infp=fopen("input.txt","rt");
    outfp=fopen("output.txt","wt");

    while( fgets(str,max_size,infp)!=NULL ) {
        fputs(str,outfp);
    }

    fclose(outfp);
    fclose(infp);
    return 0;
}
```

Standard Stream

```c
#include <stdio.h>

int main(void)
{
    char str[512];

    while( gets(str)!=NULL ) {
        puts(str);
    }

    return 0;
}
```

파일을 통한 포맷 입출력

File Stream

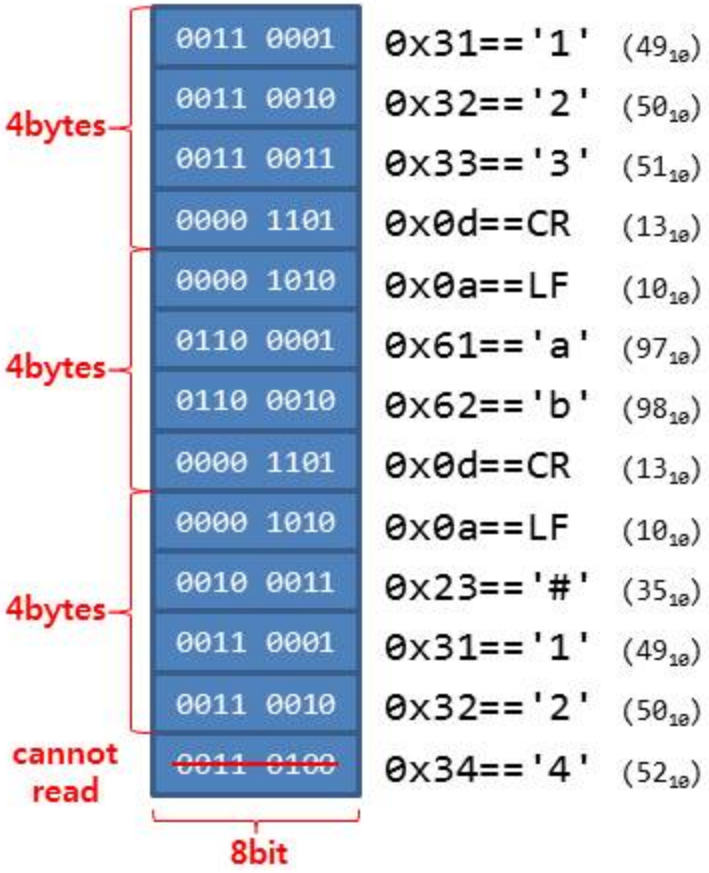
```c
#include <stdio.h>

int main(void)
{
    FILE* infp;
    FILE* outfp;
    int a;

    infp=fopen("input.txt","rt");
    outfp=fopen("output.txt","wt");

    while( fscanf(infp," %d",&a)>0 ) {
        fprintf(outfp,"%d ",a);
    }

    fclose(outfp);
    fclose(infp);
    return 0;
}
```

Standard Stream

```c
#include <stdio.h>

int main(void)
{
    int a;

    while( scanf(" %d",&a)>0 ) {
        printf("%d ",a);
    }

    return 0;
}
```

파일을 통한 이진 데이터 입출력

- Binary Data 처리 과정




```c
#include <stdio.h>

int main(void)
{
    FILE* infp;
    FILE* outfp;
    int a;

    infp=fopen("input.txt","rb");
    outfp=fopen("output.txt","wb");

    while( fread(&a,4,1,infp)>0 ) {
        fwrite(&a,4,1,outfp);
    }

    fclose(outfp);
    fclose(infp);
    return 0;
}
```