

[ cmd ]

cd ..	상위 디렉토리로 이동
file *	file의 type을 알려줌 (모든 파일)
file b*	file의 type을 알려줌 (b로 시작하는)
echo [문자열]	문자열을 출력해줌
cp f1 d1	f1 파일을 d1 디렉토리 하위에 저장
cp f1 d1/f2	f1 파일을 d1 디렉토리 하위에 f2라는 파일에 붙여넣기
rm f1	f1 제거
mv f1 f2	f1파일 이름을 f2로 변경
grep -nr "[문자열]" *	문자열을 찾아주는 명령어 -nr : 문자열 발견된 line number & 하위디렉토리 찾아가면서
ps -ef	process 설명 출력 -ef : 모든 사용자의 process
chmod	파일권한 변경 : 000 권한삭제,
gzip	파일 압축시킴
dd if=f4 of=d2 bs=512	f4 파일을 d2 디렉토에 복사, block SIZE = 512 상대참조를 위해 f4의 상위 디렉토리로 이동
df	하드디스크 용량
diff	두 개 파일을 비교(같으면 아무것도 출력 X)
dmesg	시스템 부팅메세지
du	파일, 디렉토리 용량
exit	콘솔창 종료
find	디렉토리에 저장된 파일과 디렉토리 검색
head / tail	파일의 텍스트 출력
ifconfig	네트워크 인터페이스 설정, 관련정보 조회
ln f5 f6	f6이라는 link file 생성
mount	하드웨어 장치
netstat	시스템 네트워크 연결 목록
stat	파일, 파일시스템 상태 조회
touch f7	비어있는 파일 생성
tty	터미널 이름 출력
gunzip	압축 해제
whereis	명령어 실행 파일위치,,, 등 찾아줌
which ls	특정 명령어 위치를 찾아줌

[ vi ]

o	줄바꿈과 동시에 insert모드
2yy	현재 커서로부터 2줄 복사
p	붙여넣기
무한루프 실행 시, 새로운 putty 창 연 뒤 [실행파일] &	실행되고 있는 파일이 있음에도 백그라운드로 실행
kill [PID]	PID에 맞는 파일의 진행을 중지. (ps를 통해 진행중인 파일의 PID 확인)
objdump	주어진 실행파일에 기계어를 어셈블리어로
xxd -l 20 [실행파일]	첫 20byte 까지만 출력 (16진수로)
xxd [실행파일] (매직넘버)	16진수로 출력하여 해당 파일의 정보 파악 가능

[ string ]

%s, %p	문자열 표현, 16진수로 표현
strlen(x)	x에 저장된 문자열 길이(인간이 보는 것까지)
strcpy (저장) - 배열일 때만 가능 - 포인터일 땐 '=' 사용	char *x = "hello"; //포인터 변수는 이렇게 char y[10]; // y = "hello" 는 오류! strcpy를 써야함. strcpy(y,x); //y공간에 x주소가 가르키는 문자열 저장  y = x; // Error!
strcmp (비교) - 포인터, 배열 둘다 가능 - 포인터일 때만 '==' 가능	if(strcmp(y, "hello")) == 0) //y 공간에 hello 문자열이 저장되어있다면 0을 return //x 포인터변수일 때도 가능 if (y == "hello") ==> Error if (x == "hello") // "hello"이 해당 문자열이 저장된 주소로 인식함

- scanf는 배열로 선언된 변수만 가능! //포인터 변수에 scanf를 사용하면 null이 저장.

gets(x) - 배열 일때만 가능	문자열 입력	gets()를 통해 입력받은 문자열의 길이는 정상
fgets(x, 99, stdin) - 배열 일때만 가능	(배열 일때만)	fgets()를 통해 입력받은 문자열의 길이는 '\0' 포함

fgets(x, 99, stdin) 해석 : 사용자가 입력한 값을 x에 저장하되 최대 99개의 문자만 저장.

fgets를 사용할 때 배열 마지막에 '0'을 넣어줌으로써 문자열로 마무리 지어줌

new 사용은 hw3 17번

strtok	<pre> char buf[256]; char *token[256]; fgets(buf, 255, stdin); int i=0; token[i] = strtok(buf, " "); //" "를 0으로 바꿈 for(i=1; ;i++){     token[i] = strtok(NULL, " ");     if(token[i] == NULL) break; } </pre>
--------	---

[ open, read, write ]

#include	<pre> #include &lt;fcntl.h&gt; #include &lt;sys/stat.h&gt; #include &lt;sys/types.h&gt; #include &lt;unistd.h&gt; #include &lt;stdio.h&gt;  int main(){ } </pre>
open	x = open("f1", O_RDONLY, 00777);
read	y = read(x, buf, 20);
write	<pre> char buf[50]; x1 = open("f1", O_RDONLY, 00777); x2 = open("f2", O_RDWR   O_CREAT   O_TRUNC, 00777) for(;;){     y = read(x1, buf, 20);     byte_size = byte_size + y;     if(y==0) break;     write(x2, buf, y); //buf 자리에 문자열도 가능     //buf의 값을 y byte만큼 x2에 저장 } return 0; </pre>
	0번 파일 : 키보드에서 읽어라! (~ scanf) 1번 파일 : 화면에 출력 (~printf) 2번 파일 : 화면에 에러파일

cnt 관련 나오면 hw4 15번

write	<div>파일을 받아서 출력하는(어딘가에 저장 X)</div> <pre> char buf[50]; char src[50];  scanf("%s", src);  x = open(src, O_RDONLY, 00777); for(;;){     y = read(x, buf, 1);     if(y==0) break;     write(1, buf, y); //buf 자리에 문자열도 가능 } return 0; </pre>
write	<div>파일 이름 직접 입력</div> <pre> char buf[50]; char src[50]; char dest[50];  scanf("%s", src); scanf("%s", dest);  x1 = open(src, O_RDONLY, 00777); x2 = open(dest, O_RDWR   O_CREAT   O_TRUNC, 00777) for(;;){     y = read(x1, buf, 20);     byte_size = byte_size + y;     if(y==0) break;     write(x2, buf, y); //buf 자리에 문자열도 가능 } return 0; </pre>

int y = 10

write(x, &y, 4);      --> 했을 때 10의 ASCII 코드가 출력됨.

[ WAV ]

- 1) wav 파일을 xxd로 표현하면 앞에 44bytes는 head 내용이 들어감
- 2) 문자열은 read를 하고 값을 파일에 저장할 때, 마지막 index에 0을 넣어줘야함  
숫자는 read를 하고 값을 저장할 파일에 &를 붙여줌.
- 3) 값을 출력이 아닌 저장  
FILE \*fout = open("sw2-wav.txt", "w"); //텍스트 파일을 FILE을 통해 open  
fprintf()를 통해 값을 저장하는데, fout을 아규먼트로 사용
- 4) wav 파일에서 lseek(x, 44, SEEK\_SET);을 통해 44번째 byte로 이동하여 이후 값을 읽는다. 전체 사이즈(30260)의 절반만큼 반복한다. l를 15108만큼 반복
- 5) wav 파일 음성 수정

[ command line argument ]

```
void main(int argc, char *argv[]){

}
```

아규먼트 몇 개든 받기	<pre>void main(int argc, char *argv[]){     int i;     for(i=0; i&lt;argc; i++){         printf("");     } }</pre>
myecho.c - 아규먼트 (문자열)의 내용 출력	<pre>void main(int argc, char *argv[]){     int i;     for(i=0; i&lt;argc; i++){         printf("%s ", argv[i]);     } }</pre>
mycat.c - 아규먼트 (파일)의 내용 출력	<pre>void main(int argc, char *argv[]){     int x, y;     char buf[20];     x = open(argv[1], O_RDONLY, 00777);     if (x== -1){         perror("error in open");         exit(1);     }     for(;;){         y = read(x, buf, 20);         if(y==0) break;         write(1, buf, y);     } }</pre>

8), 9),	함수형태로 묶기
입력된 모든 아규먼트 (파일) 내용 출력	<pre> void main(int argc, char *argv[]){     int x, y;     char buf[20];     int i;     for(i=1;i&lt;argc;i++){         x = open(argv[i], O_RDONLY, 00777);         if (x==-1){             perror("error in open");             exit(1);         }         for(;;){             y = read(x, buf, 20);             if(y==0) break;             write(1, buf, y);         }     } } </pre>
mycp.c	<pre> void main(int argc, char *argv[]){     int x, y;     char buf[20];     x1 = open(argv[1], O_RDONLY, 00777);     x2 = open(argv[2], O_RDWR   O_CREAT   O_TRUNC, 00777)     if (x1==-1    x2 == -1){         perror("error in open");         exit(1);     }     for(;;){         y = read(x1, buf, 20);         if(y==0) break;         write(x2, buf, y);     } } </pre>

myxxd.c	<pre> void main(int argc, char *argv[]){     int x, y;     char buf[20];     x = open(argv[1], O_RDONLY, 00777);     if (x== -1){         perror("error in open");         exit(1);     }     for(;;){         y = read(x, buf, 1);         if(y==0) break;         printf("%x ", buf[0]);     } } </pre>
옵션 만들기	<pre> #include &lt;fcntl.h&gt; #include &lt;sys/stat.h&gt; #include &lt;sys/types.h&gt; #include &lt;unistd.h&gt; #include &lt;stdlib.h&gt; #include &lt;stdio.h&gt; #include &lt;string.h&gt; #include &lt;dirent.h&gt; </pre>