

make

make?

- make
 - GNU make utility to maintain groups of programs
 - 프로그램 그룹을 유지하는데 필요한 유틸리티
 - 프로그램 그룹 중 새롭게 컴파일되어야 하는 부분을 자동적으로 판단
 - 필요한 명령을 이용 컴파일 진행
 - 프로그램 개발(컴파일 작업 등) 뿐 아니라 명령어 방식으로 처리되는 모든 곳에 쓰임(batch)

- 장점
 - 입력파일 변경시 결과파일 자동 변경을 원할때
 - 지능적인 배치작업 수행
- Makefile
 - make 작업의 기본이 되는 쉘 스크립트
 - 결과 파일을 생성시키기 위한 파일의 연결, 참조, 명령 등을 기록

make 필요성

- 프로그램 개발 및 유지가 주 목적
 - 프로그램 컴파일 시 컴파일 양이 많을 때 파일을 나누어 컴파일 시도
 - 연관관계를 갖는 파일들간에 하나의 파일이 변경되었을 때 연관되는 다른 파일도 같이 변경되어 컴파일 시도
 - 파일 수가 많아 일일이 컴파일을 시도하지 못할때 전체 파일의 컴파일 시도
 - 컴파일시 지능적으로 새로운 정보를 갖는 파일만 컴파일 시도

Makefile

- Makefile 구성
 - 목적파일(Target)
 - 의존성(Dependency)
 - 명령어(Command)

```
target : dependency ...  
        command
```

...

...

목적파일(target)은 명령어
(command)가 수행되어 나온 결과
파일(Object, Excute file)을 지정

명령어는 의존관계에 정의된 파일
의 내용이 바뀌거나 목적파일이
없을 때 이 부분의 명령어가 차례
로 실행됨

- Makefile 작성시 참고사항

- target 부분에는 결과 파일 뿐 아니라 간단한 레이블(label)을 기록할 수 있음
- 명령어 부분은 반드시 TAB으로 시작
- make 실행시 에러가 날 때
TAB으로 시작되지 않는 명령어 구문이 대부분

Makefile 예제

- Makefile 작성

- 작업 수행 (직접 입력 명령어)

- # gcc -c main.c

- # gcc -c read.c

- # gcc -c write.c

- ...

- # gcc -o test main.o read.o write.o ...

- Makefile 예제

```
test : main.o read.o write.o
```

```
    gcc -o test main.o read.o write.o
```

```
main.o : io.h main.c
```

```
    gcc -c main.c
```

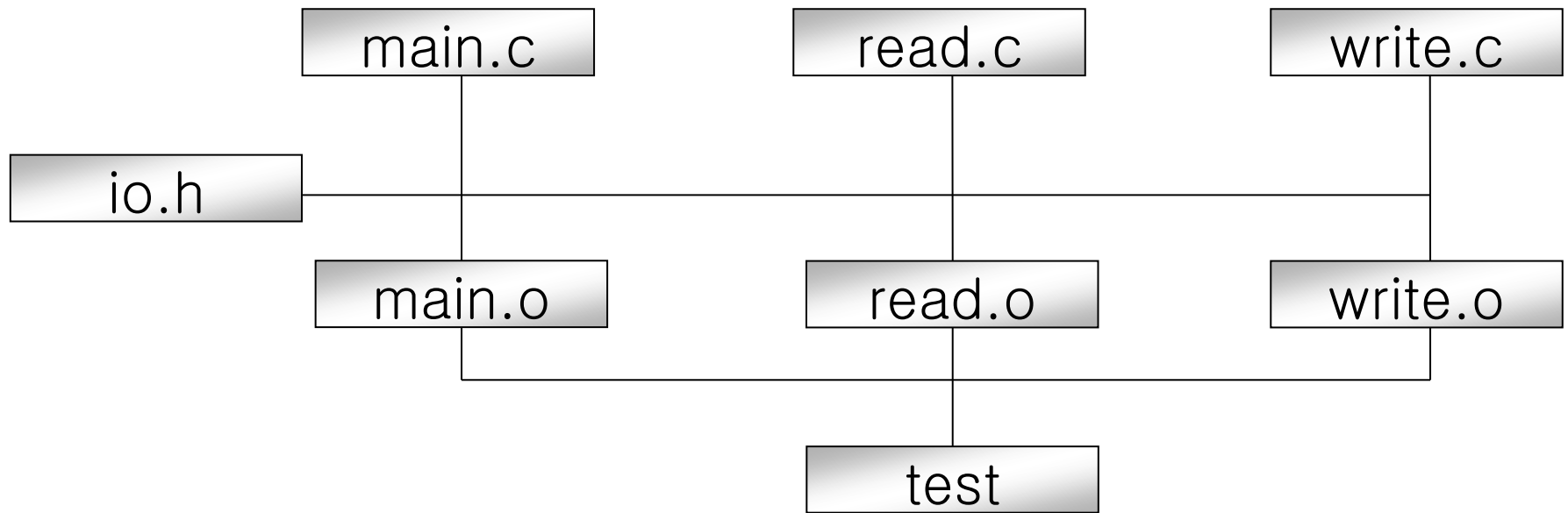
```
read.o : io.h read.c
```

```
    gcc -c read.c
```

```
write.o : io.h write.c
```

```
    gcc -c write.c
```


Makefile 예제 연결



test가 생성되기 까지 과정

c 파일은 io.h 파일에 의존하여 obj 파일을 생성하고
생성된 obj 파일을 링크하여 test가 생성

- Makefile 예제 실행 순서

```
# make
```

```
gcc -c main.c
```

```
gcc -c read.c
```

```
gcc -c write.c
```

```
gcc -o test main.o read.o write.o
```

매크로의 사용

- 매크로
 - 변수를 이용하여 쉘 프로그램을 간단하게 수정
 - 반드시 변수는 \$() 형식으로 사용

- 예) Makefile 예제 2

```
OBJECTS = main.o read.o write.o
```

```
test : $(OBJECTS)
```

```
    gcc -o test $(OBJECTS)
```

```
main.o : io.h main.c
```

```
    gcc -c main.c
```

```
...
```

레이블(Label)의 사용

- 레이블(Label)
 - Target 부분을 레이블로 사용 가능

- 예) Makefile 예제 3

```
OBJECTS = main.o read.o write.o
```

```
test : $(OBJECTS)
```

```
    gcc -o test $(OBJECTS)
```

```
main.o : io.h main.c
```

```
    gcc -c main.c
```

```
read.o : io.h read.c
```

```
    gcc -c read.c
```

```
write.o: io.h write.c
```

```
    gcc -c write.c
```

```
clean :
```

```
    rm $(OBJECTS)
```

- 실행...

```
# make
```

```
...
```

```
# make clean
```

```
rm main.o read.o write.o
```