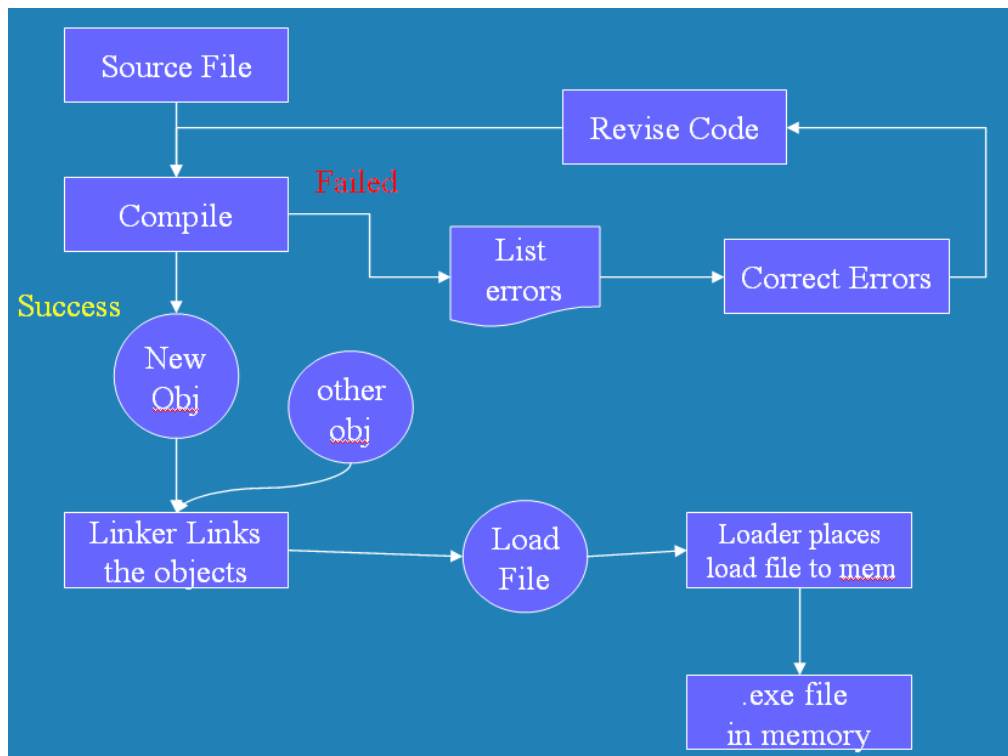


- make is a UNIX utility for building projects that are comprised of multiple source files
- make takes care of dependencies, and
- rebuild the project if one or more file is modified
- a file, usually named makefile, contains details of the dependencies, and therefore tells make how to build the project
- Before we go to details and sample make file it will be beneficial to know about how .exe files are created by compilers



Demo: dttime.h, dttime.cpp timedemo.cpp

```
g++ -c dttime.cpp
```

```
g++ -c timedemo.cpp
```

```
g++ dttime.o timedemo.o -o timedemo
```

Dependency Notation

- **Target depends on source. If source has changed, execute the command**

```
target: source [source2]
```

```
command
```

```
[command]
```

For example, a makefile to build the executable file one from the C/C++ source files one.cpp, two.cpp and three.cpp

```
one      one.o one.cpp      // Dependencies
```

```

        one.h
two      two.o two.cpp
        two.h
three    three.o three.cpp
        three.h

```

```

one: one.o two.o three.o
    g++ -o one one.o two.o three.o

```

```

one.o: one.cpp one.h
    g++ -c one.cpp

```

```

two.o: two.c two.h
    g++ -c two.cpp

```

```

three.o: three.cpp three.h
    g++ -c three.cpp

```

There are no source files specified, execute command unconditionally:

```

target:
    command
    [command]

```

Example

```

clean:
    rm -rf timedemo.o dttime.o timedemo

```

Demonstration:

see home/comp2710/makefile
 see home/comp2710/makefile_v2
 see home/comp2710/makefile_better

Demo: dttime.h, dttime.cpp timedemo.cpp

```

g++ -c dttime.cpp
g++ -c timedemo.cpp
g++ dttime.o timedemo.o -o timedemo

```

```

timedemo: dttime.o timedemo.o
    g++ dttime.o timedemo.o -o timedemo

```

Which files does dttime.o depend on?

```

dttime.o: dttime.cpp dttime.h

```

```
g++ -c dtime.cpp
Which files does timedemo.o depend on?
timedemo.o: timedemo.cpp
g++ -c timedemo.cpp
```

Exercise: See [home/comp2710/separate-compilation](#)

```
#Try the variable CC
CC = g++

#Let's try a flag
CFLAGS = -c
#all: estimate_demo

estimate_demo: estimate_function.o estimate.o
$(CC) estimate_function.o estimate.o -o estimate_demo

estimate.o: estimate.cpp estimate_function.h
$(CC) $(CFLAGS) estimate.cpp

estimate_function.o: estimate_function.cpp estimate_function.h
$(CC) $(CFLAGS) estimate_function.cpp

clean:
rm -rf estimate_function.o estimate.o estimate_demo
```