A Programming Problem

Consider a programming project consisting of:

1. Three C source files: main.c, iodat.c, dorun.c
2. An assembly routine: lo.s
3. Library routines is /usr/fred/lib/crtn.a

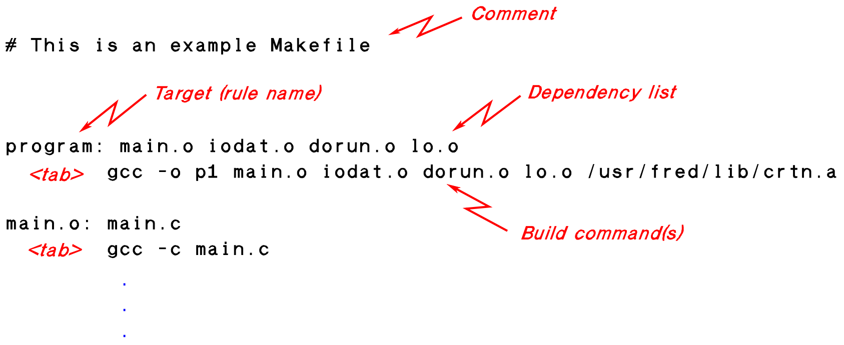
To Compile Each Separately:

```
gcc -c main.c

gcc -c iodat.c

gcc -c dorun.c

as -o lo.o lo.s

gcc -o p1 main.o iodat.o dorun.o lo.o /usr/fred/lib/crtn.a
```

A makefile:

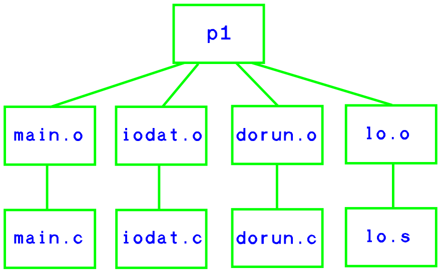
```
# This is an example Makefile

program: main.o iodat.o dorun.o lo.o
        gcc -o p1 main.o iodat.o dorun.o lo.o /usr/fred/lib/crtn.a

main.o: main.c
        gcc -c main.c

iodat.o: iodat.c
        gcc -c iodat.c

dorun.0: dorun.c
        gcc -c dorun.c

lo.o: lo.s
        as -o lo.o lo.s
```

Dissecting the Makefile:



```
# This is an example Makefile

program: main.o iodat.o dorun.o lo.o
  <tab>  gcc -o p1 main.o iodat.o dorun.o lo.o /usr/fred/lib/crtn.a

main.o: main.c
  <tab>  gcc -c main.c
         .
         .
         .
```

Dependence Graph



p1

main.o iodat.o dorun.o lo.o

main.c iodat.c dorun.c lo.s

make uses "backward-chaining" to build the dependence graph.

Another makefile Example

```
all: plot_prompt plot_win

plot_prompt: basic.o prompt.o
        gcc -o plot_prompt basic.o prompt.o

plot_win: basic.o window.o
        gcc -o plot_win basic.o window.o

basic.o: basic.c basic.h
        gcc -c basic.c

prompt.o: prompt.c basic.h
        gcc -c prompt.c

window.o: window.c
        gcc -c window.c
clean:
        rm -f *.o core plot_prompt plot_win
```

Macros in makefiles

```
LIBES = -lX11
objs = drawable.o plotpts.o rdata.o
CC = /usr/fred/bin/cc
CFLAGS =   # can put compiler options here later
BINDIR = /usr/local/bin

plot: ${objs}
        ${CC} -o plot ${CFLAGS} ${OBJS} ${LIBES}
        mv plot ${BINDIR}
        .
        .
        .
```

When executed:

```
make plot
/usr/fred/bin/cc -o plot drawable.o plotpts.o rdata.o -lX11
mv plot /usr/local/bin
```