MAKEFILES

Build Process

- **Compiler** takes source files (.c) and outputs object files (.o)

- **Linker** takes the object files and creates an executable

Example Program

- A program is made of 5 files:
  - main.c, a main program
  - point.h, header file for Point
  - point.c, implementation file for Point
  - rectangle.h, header file for Rectangle
  - rectangle.c, implementation file for Rectangle

- Assuming that these are all in the same directory, how would you **compile** the above in the shell?
- What files are generated after compilation?
- How would you **link** the files to create the executable?

Dependency Chart

- Used to determine which of the files need to be **regenerated** when we change a part of the program

- For example, which commands do I need to run if I modified:
  - main.c?
  - rectangle.c ?

- Let’s build a dependency chart for the program on the previous slide

Using Dependency Chart

- Suppose we change the file main.c. What needs to be regenerated?

Makefile

- Makefiles consist of:
  - A set of variables
  - A set of targets to be generated
- Anything that starts with a # is a comment

Makefile (complete it)

```
# Makefile for Writing Make Files Example

# ****************************************************
# Variables to control Makefile operation

CC = gcc
CFLAGS = -Wall -g

# ****************************************************
# Targets needed to bring the executable up to date

main: main.o point.o rectangle.o
	$(CC) $(CFLAGS) -o main main.o point.o rectangle.o

main.o: main.c point.h rectangle.h
	$(CC) $(CFLAGS) -c main.c

point.o:

rectangle.o:
```

Makefile

- In a make file, if you need to continue a line, you cannot just continue it on the next line
- You must end a line with a \ (backslash, not the forward slash) to tell make that the line continues
- Only break lines where space would normally go

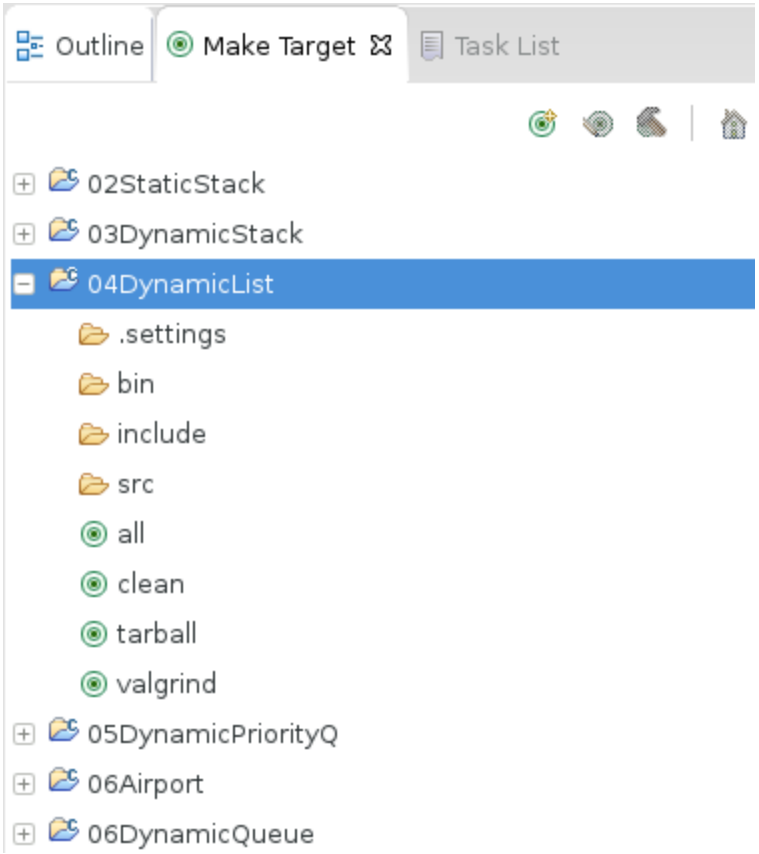
Example with Linebreaks

```
main: main.o \
      point.o \
      rectangle.o
     $(CC) $(CFLAGS) -o main \
           main.o point.o rectangle.o
```

Make Targets

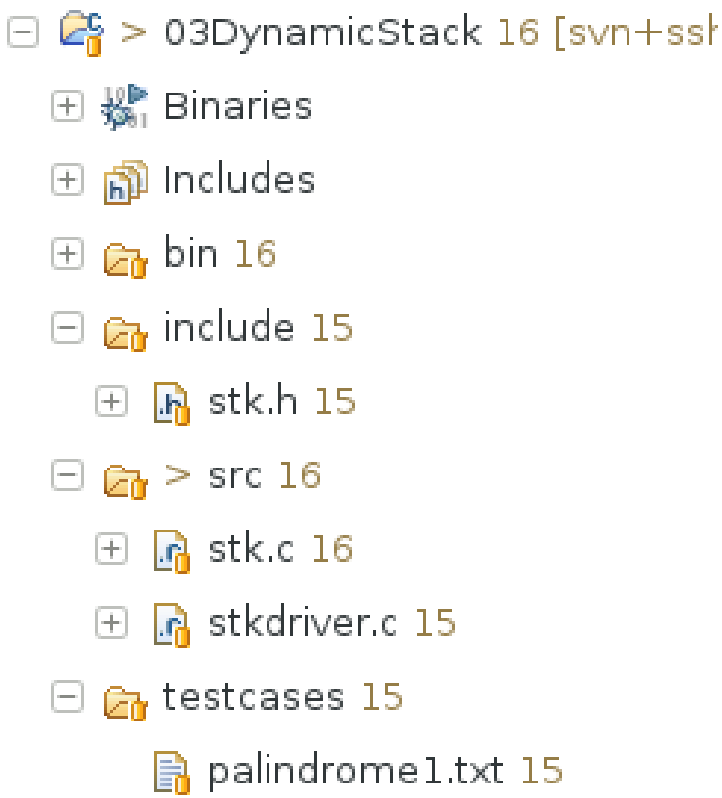
- You can create shortcuts to all of your make targets in Eclipse
  - Window->Show View->Make Target
- Right click on a project and select (Create Make Target), then type in the target name (for example: all, clean)

Make Target


Outline
Make Target
Task List
02StaticStack
03DynamicStack
04DynamicList
.settings
bin
include
src
all
clean
tarball
valgrind
05DynamicPriorityQ
06Airport
06DynamicQueue

Your Turn

- Create a Makefile for the following project:



Other Targets

- clean
- valgrind
- tarball

Modify Makefile

- Modify the previous Makefile to include a new target for palindromeChecker