COMP20200 Unix Programming

Lecture 09

CS, University College Dublin, Ireland

main.c

```c
#include <stdio.h>
double area(int r);
int main (void) {
  double radius = 10.0;
  double ans = area(radius);
  printf ("Circle r:%2.2lf a: %2.2lf\n", radius, ans);
  return 0; }
```

area.c

```c
#define PI 3.14
double area(int r){
  return PI * r * r; }
```

Compile with:

```
gcc -Wall -o circle main.c area.c
```

main.c

```
#include <stdio.h>
double area(int r);
int main (void) {
   ...
   }
```

area.c

```
double area(int r){
   ...
}
```

What if we redefine function as:

```
double area(double r);
```

Need to fix everywhere area is used, mistakes could happen.
Use header files instead.

circle.h

```
#define PI 3.14
double area(double r);
```

main.c

```
#include <stdio.h>
#include "circle.h"
int main (void) {
   ...
   }
```

area.c

```
#include "circle.h"
double area(double r){
   ...
}
```

Still compile with:

```
gcc -Wall -o circle main.c area.c
```

Not: `gcc ... circle.h ...`

Makefile

- Automate building of program
- Ideal when there is lots of source files.
- Only recompile each source file when needed.

```
CC=gcc
CFLAGS=-g -Wall
LIBS=-lm

circle: main.o area.o
	$(CC) $(LIBS) main.o area.o -o circle

main.o: main.c
	$(CC) -c $(CFLAGS) main.c

area.o: area.c
	$(CC) -c $(CFLAGS) area.c
```

Makefile

- Makefile basic syntax:

```
target: dependencies
[tab] system command
```

- Eg: Default make target `all`

```
all:
    gcc -g -Wall -o circle main.c area.c -l.
```

- Then compile package with:

```
$ make all
$
```

Makefile continued

```
# comment start with #
CC=gcc
# options passed to compiler:
CFLAGS=-g -Wall
LIBS=-lm

all: circle

circle: main.o area.o
	$(CC) $(LIBS) main.o area.o -o circle

main.o: main.c
	$(CC) -c $(CFLAGS) main.c

area.o: area.c
	$(CC) -c $(CFLAGS) area.c

clean:
	rm -rf *.o circle
```

Makefile continued

```
CC=gcc
CFLAGS=-g -Wall
LIBS=-lm

all: circle

circle: main.o area.o
	$(CC) $(LIBS) main.o area.o -o circle

main.o: main.c
	$(CC) -c $(CFLAGS) main.c

area.o: area.c
	$(CC) -c $(CFLAGS) area.c

main.c: circle.h
area.c: circle.h

clean:
	rm -rf *.o circle
```

Makefile continued

```
CC=gcc
CFLAGS=-g -Wall
LIBS=-lm
HFILES=circle.h
OBJ=main.o area.o

all: circle

circle: $(OBJ)
	$(CC) $(LIBS) $^ -o $@

%.o: %.c $(HFILES)
	$(CC) -c $(CFLAGS) $<

clean:
	rm -rf *.o circle
```

Makefile

- Can set variables at command line:

```
$ make CFLAGS="-g -O2"
```