CS221
C and Systems Programming

Basic Makefile

```
target: dep1.c dep2.c
    gcc –o target dep1.c dep2.c
```

Save compile time by saving intermediate results

```
target: dep1.o dep2.o
    gcc –o target dep1.o dep2.o
dep1.o: dep1.c
    gcc –c dep1.c –o dep1.o
dep2.o: dep2.c
    gcc –c dep2.c –o dep2.o
```


Only recompile the file(s) that changed.

Use VARIABLES for convenience

```
GCC = gcc -g
OBJS = dep1.o dep2.o
target: $(OBJS)
    $(GCC) -o target $(OBJS)
dep1.o: dep1.c
    $(GCC) -c dep1.c -o dep1.o
dep2.o: dep2.c
    $(GCC) -c dep2.c -o dep2.o
```

Built-in VARIABLES

```
GCC = gcc -g
OBJS = dep1.o dep2.o
target: $(OBJS)
    $(GCC) -o $@ $(OBJS)
dep1.o: dep1.c
    $(GCC) -c $< -o $@
dep2.o: dep2.c
    $(GCC) -c $< -o $@
```


$@: target
$<: dependency

Use "pattern rules"

```
GCC = gcc -g
OBJS = dep1.o dep2.o
target: $(OBJS)
    $(GCC) -o $@ $(OBJS)
$(OBJS): $(OBJS:%.o=%.c)
    $(GCC) -c $< -o $@
```

Use "pattern rules" Part 2

```
GCC = gcc -g
OBJS = dep1.o dep2.o
target: $(OBJS)
    $(GCC) -o $@ $(OBJS)
$(OBJS): %.o: %.c
    $(GCC) -c $< -o $@
```


Only rebuild the objects that need to be rebuilt!

*.o files in subdirectory!

```
GCC = gcc -g
OBJS = dep1.o dep2.o
SUBDIR = obj
TOBJS = $(OBJS:%=$(SUBDIR)/%)
target: $(TOBJS)
    $(GCC) -o $@ $(TOBJS)
$(TOBJS): $(SUBDIR)/%.o: %.c
    $(GCC) -c $< -o $@
```

"fake" targets

```
GCC = gcc -g
OBJS = dep1.o dep2.o
all: target
clean:
    /bin/rm -f $(OBJS) target
git:
     git add $(OBJS:%.o=%.c)
     git commit -m "Git commit from Make"
     git push
…
```