

Summary Page: Make

CISC 220, fall 2012

Command-line arguments:

- f <filename> means to use <filename> as the "make file", containing rules and dependencies. Without -f, make looks for a file called makefile or Makefile.
 - n: don't actually make the target, just print the commands that make would execute to make the target
 - k: keep going in spite of errors (default behavior is to stop if a command fails)
 - B: assume all files have changed (rebuild the target from scratch)
 - <var>=<value>: sets the value of a make variable, overriding its definition in the make file, if any. For example, make CFLAGS='-ansi -Wall'
- Specifying a target: An argument that is not a flag starting with "-" or a variable assignment is a target to be made. For example, make lab5.o makes the lab5.o file.
- If you don't specify a target, uses the first target in the make file.

Rule Syntax:

```
target ... : prerequisites ...
               command
               ..
```

or:

```
targets : prerequisites ; command
```

In multi-line syntax, command lines must start with a tab. Beware of editor settings that replace tabs with spaces!

Variables:

To define and set a variable in a make file:

```
<varname> = <value>
```

To use the variable:

```
$(<varname>)
```

or:

```
${<varname>}
```