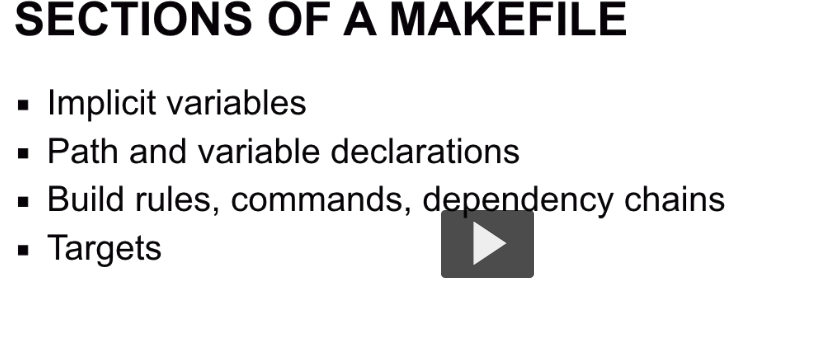
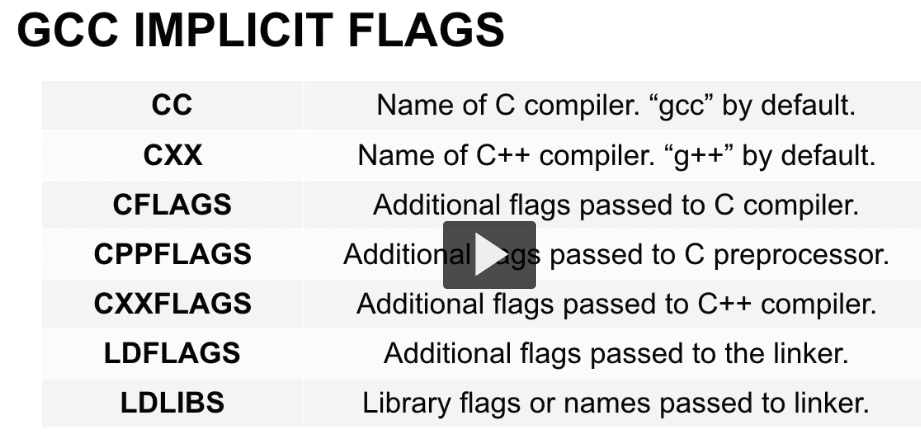
**Makefile Demo’s:**



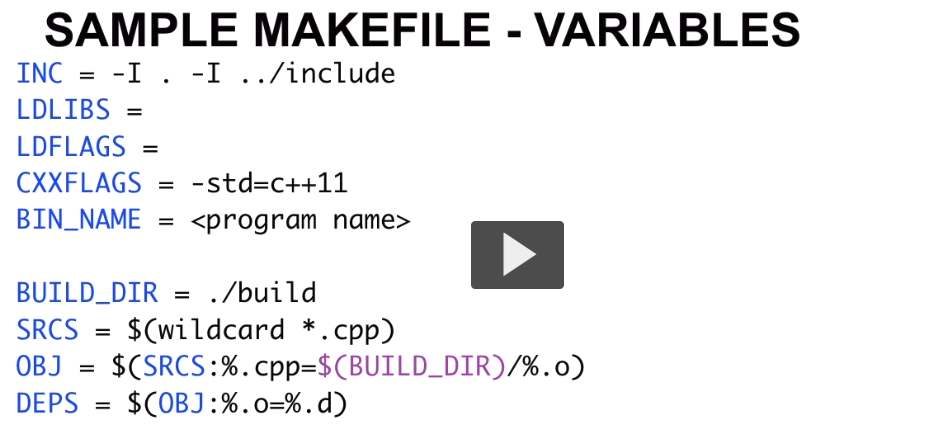
**1.Implicit variables within the makefile**



**Usually makefile contains 3 targets all clean and test**

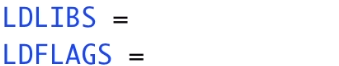


Implicit **Variables that are available in make that we don’t have to specify or we override and provide additional values**



1.We have Inc to include directory

I choose this name arbitrary you can use name anything you want

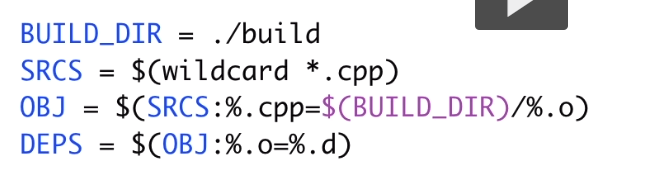




Adding c++11



You see that is used later

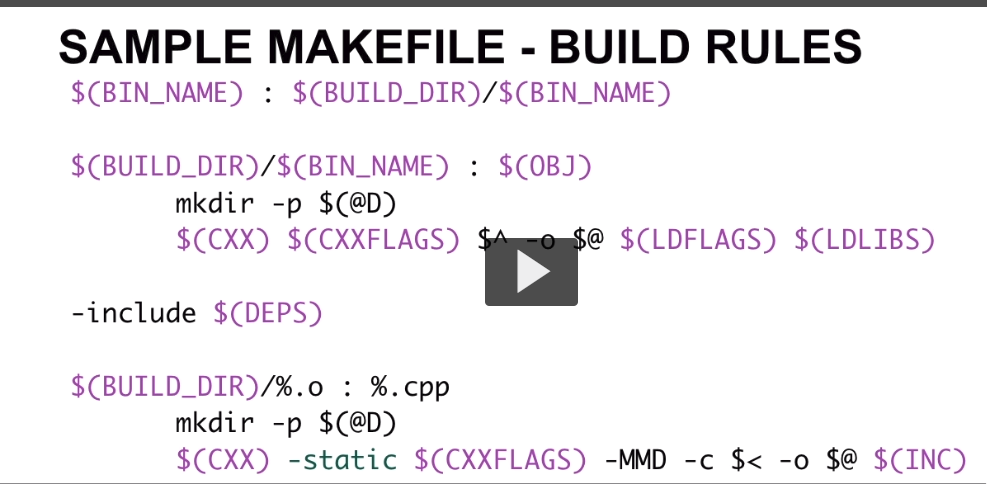


Obj stands for the intermediate files

All of the cpp files that get turned into o files

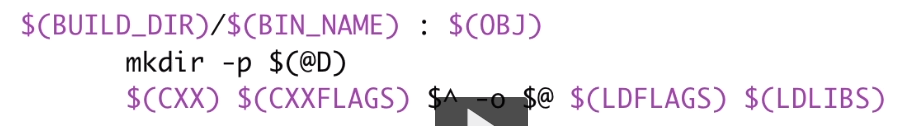


Is for dependency that are created from all of the .0 files that are automatically generated into .d files





Bin name is target name depends upon build directory and build name existing



This section is depend upon object files and if object files don’t exist they it will rebuild

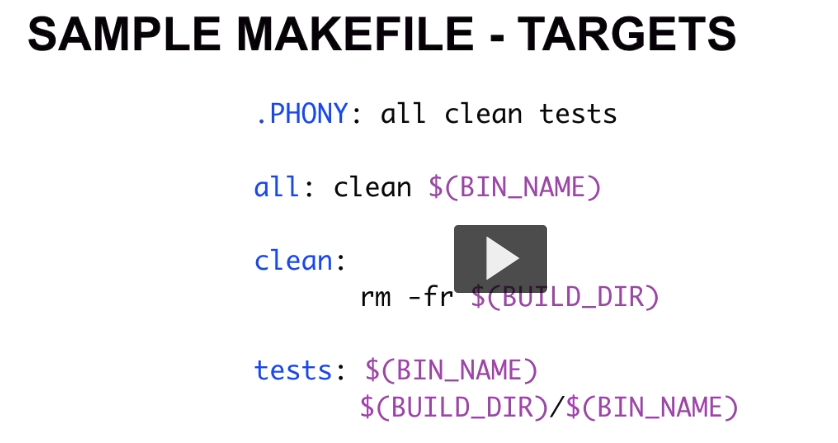
Makefile is list of stack dependies



Make a dir for builder



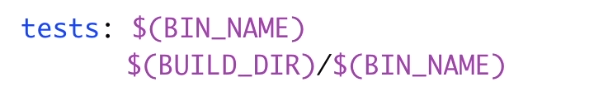
C++ compiler that will create output file



..Phoney is doing 3 targets 1.all 2.clean 3.tests

All dependent upon clean and $(BIN\_NAME)

First it run clean target and then run $Bin name



Now all make files are simple text files and instead of spaces

All white spaces must be accomplished by using the tab character.

So anymake file that contains single space for white space that invalid makefile



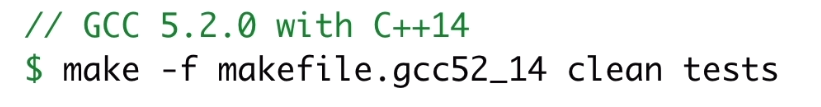
Run the clean target first then execute the test target



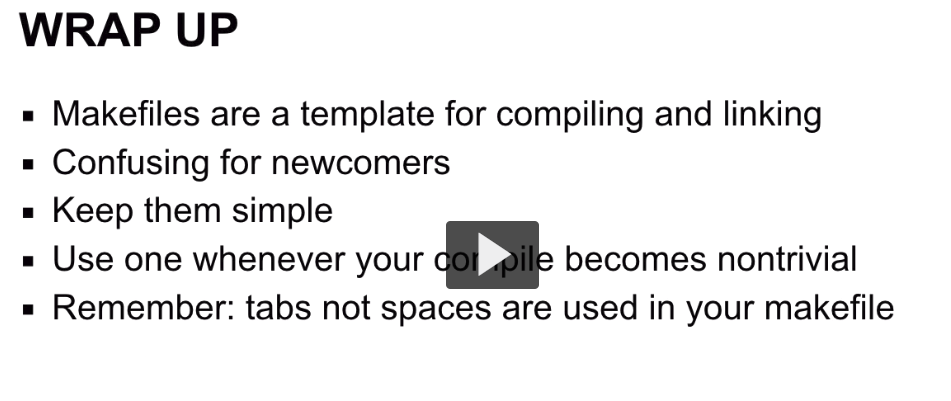
1.lower case named makefile it is picked

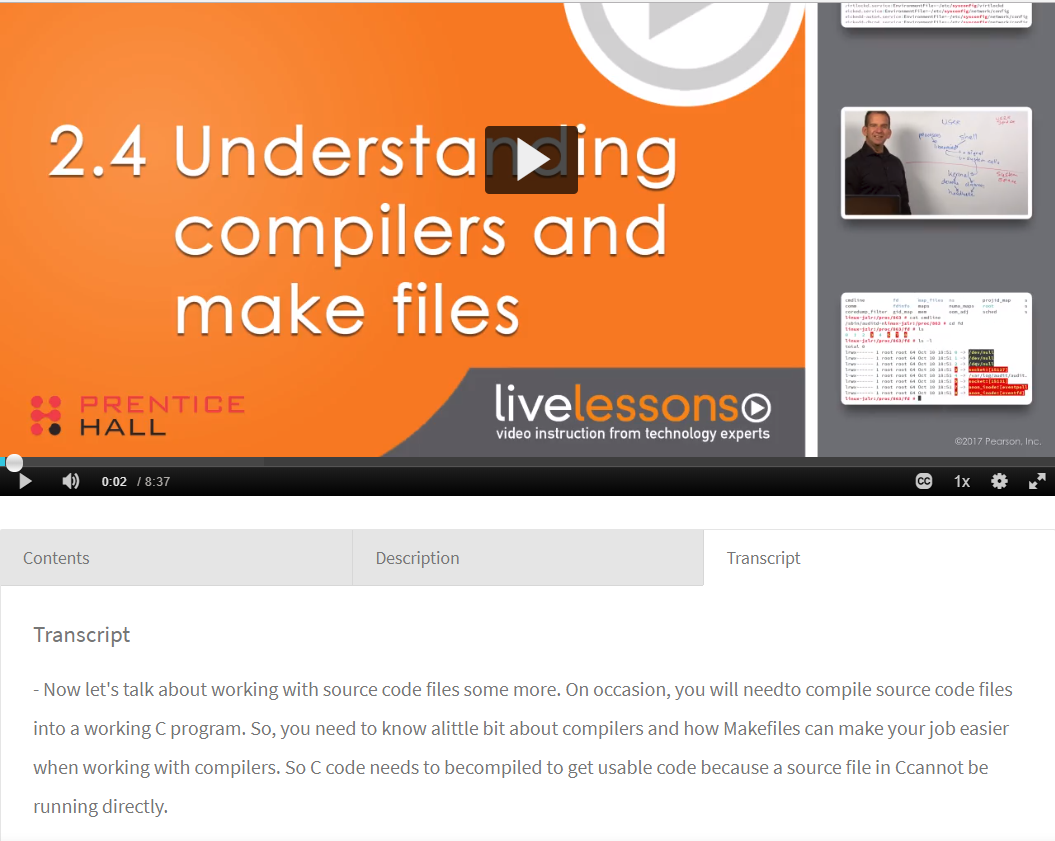
2.if it named uppercase MAKEFILE then it wouldn’t update

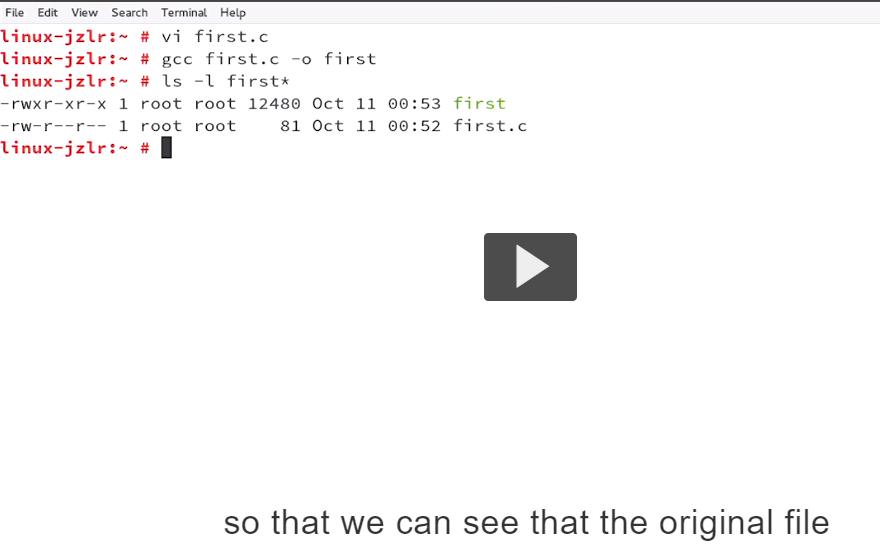
3.If it named Makefile like that then it is picked up



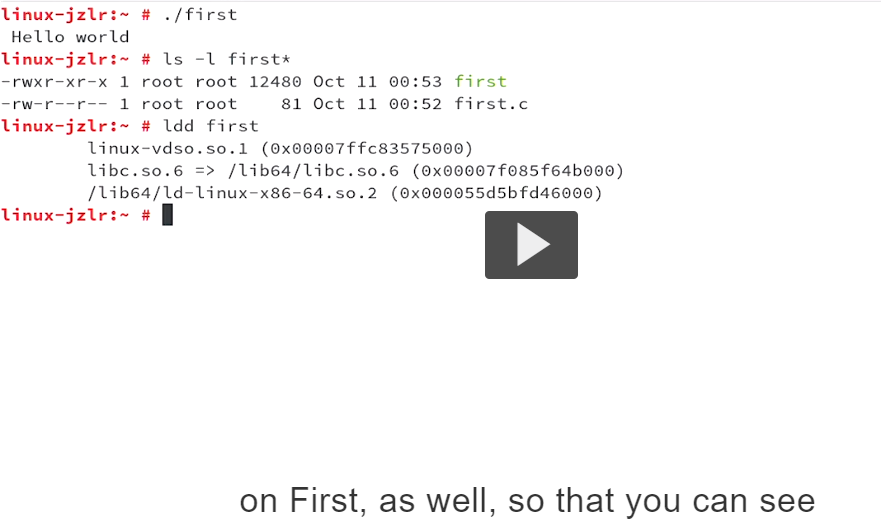
Use –f for custom makefile name like makefile.gcc52\_14

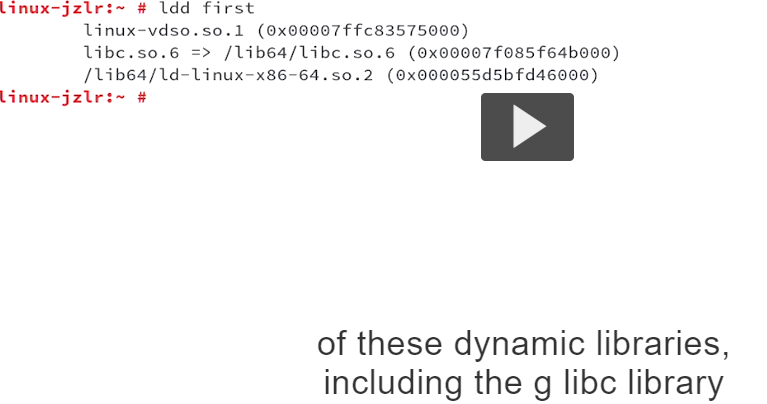


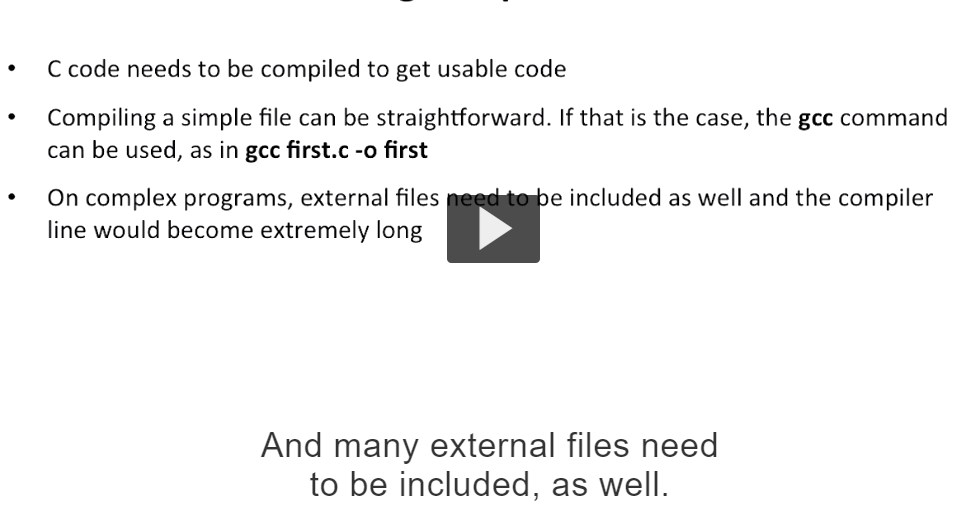


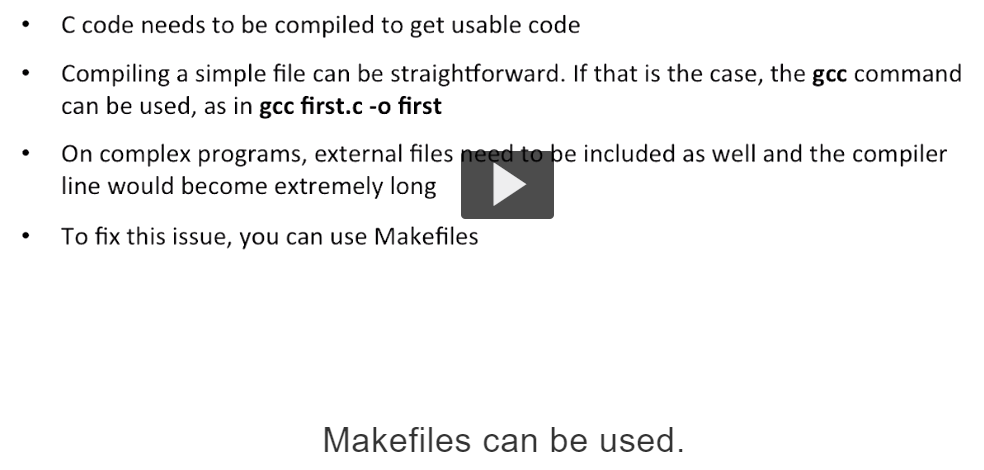


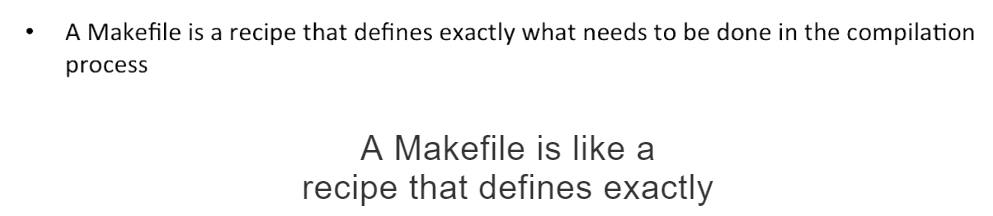


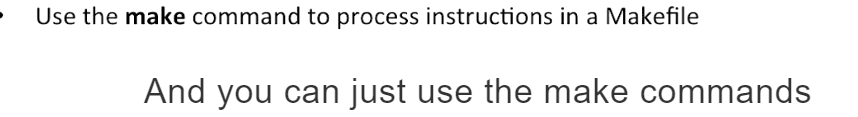


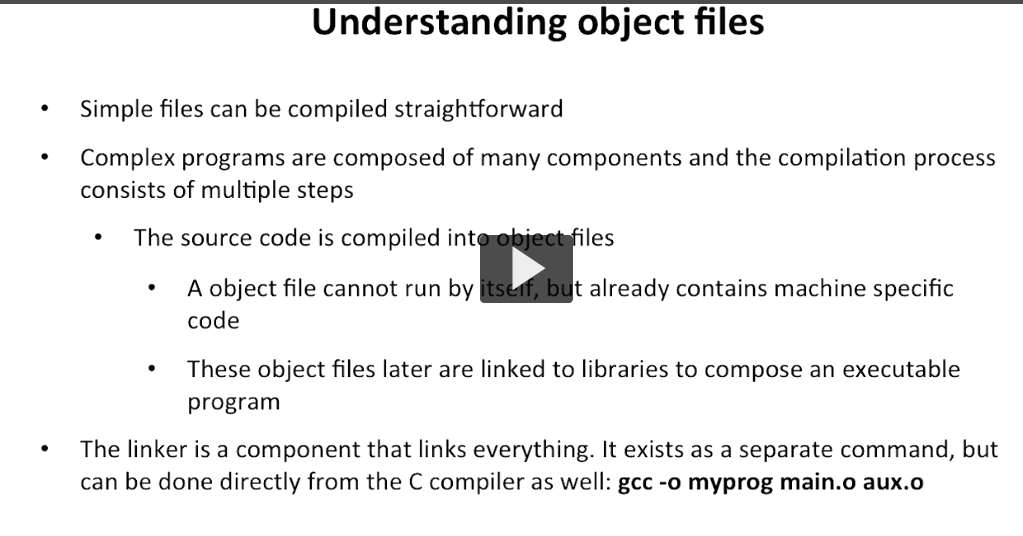




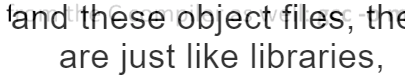


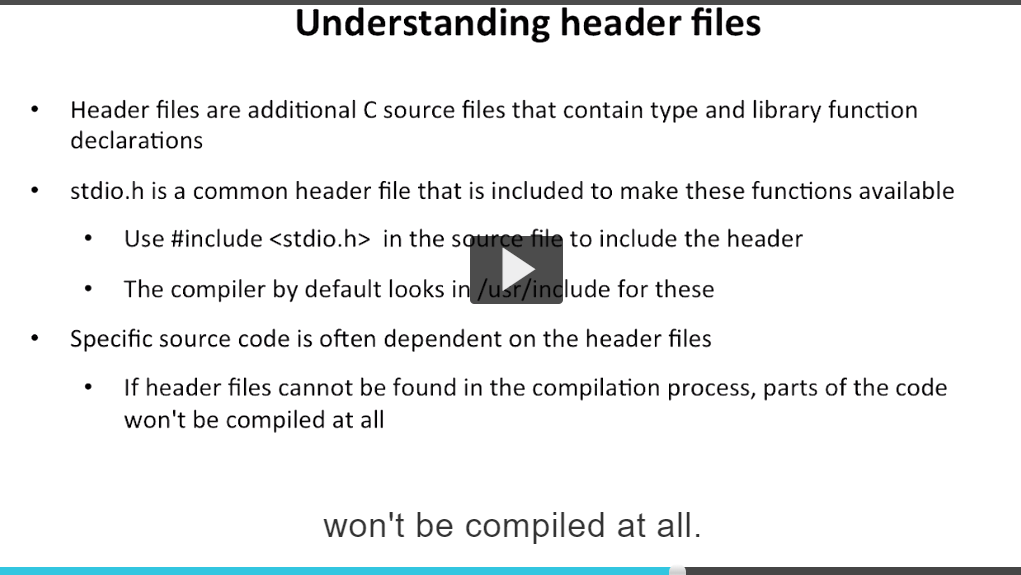


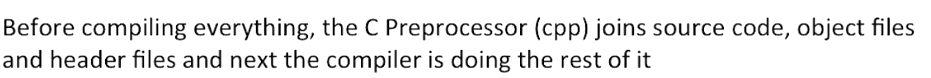


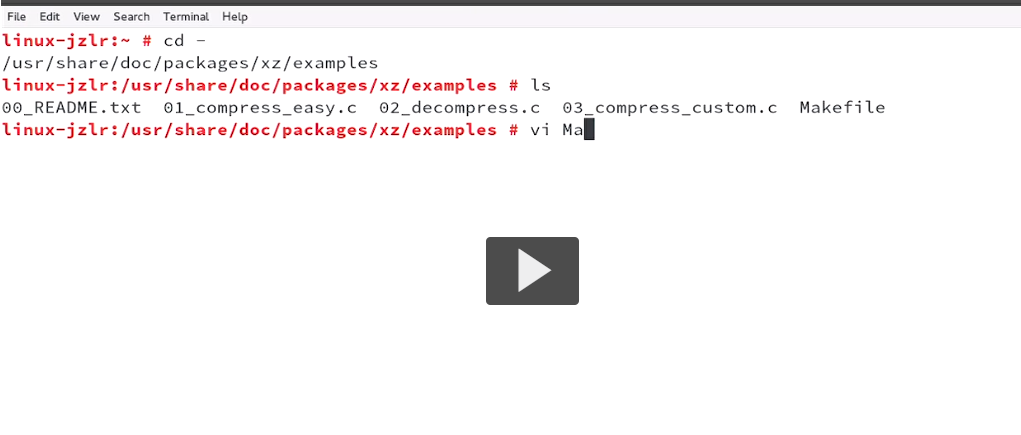


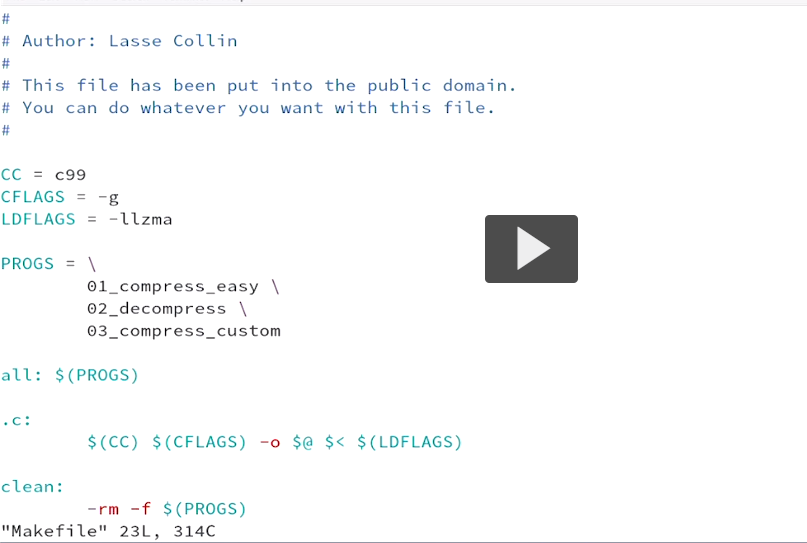














Compiler

