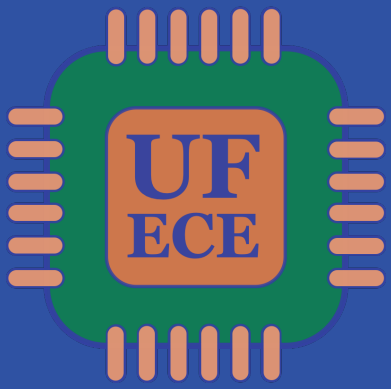
Microprocessor Applications

"C" Programming



UF UNIVERSITY of FLORIDA

Dr. Eric Schwartz
Christopher Crary
Wesley Piard

1

2

Lecture Outline

The following will be covered within this lecture:

- Motivation for a programming language other than an assembly or machine language.
- A brief introduction to the "C" programming language.
- A listing of some useful resources for learning how to apply the "C" programming language.

NOTE: This lecture is not a tutorial on how to utilize the "C" programming language; such instruction should be sought throughout other content.

3

Motivation for Other Forms of Programming

- Although assembly and machine languages provide a programmer with some means for directly controlling the operation of a microprocessor, writing a computer program with such languages often requires a significant amount of effort.
- Overall, this is because these languages are often notably different that that of a natural human language.
- Thus, so that writing computer programs may be more efficient and effective, it would be very useful to have the ability to instruct a computing system via some more abstract, closer-to-human language.

3

4

Motivation for Other Forms of Programming, cont.

- Fortunately, in the modern world, there already exist many computer programming languages (several hundred, in fact) that allow humans to communicate with computing systems in a more productive manner.
- Out of all computer programming languages that exist today, one of the most mature, well-established ones is the “C” programming language.

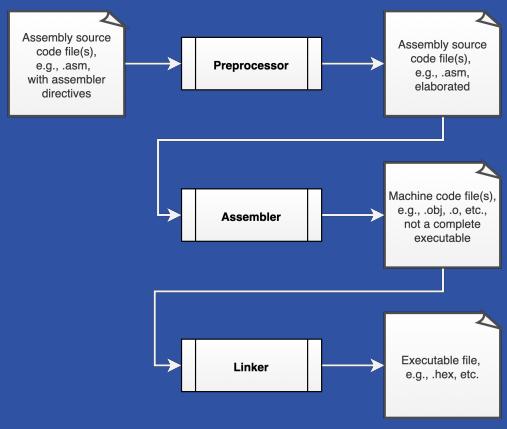
4

Introducing the "C" Programming Language

- Overall, "C" is a simple, general-purpose programming language meant provide a programmer with almost as much control as an assembly language.
- In fact, this is its main advantage over most other programming languages. Other languages generally sacrifice a higher level of control for a more abstract, or more specialized, grammar.
  - There exists a separate programming language "C++" that provides, in effect, the same benefits as "C" and more, although it is not yet (but is close to being) as widely supported as "C".
- Because of its powerful nature, and simply because it is so widely supported by most computing system platforms, knowledge of how to utilize the "C" programming language is invaluable to most any computer programmer.

Previous Programming "Toolchain" Model

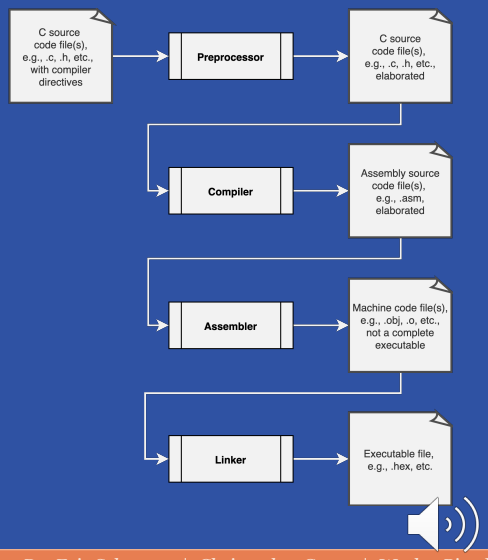
- Provided on the right is a partial depiction of the generic programming toolchain model we have utilized throughout this course thus far.
  - Application code has been written with an assembly language, incorporating preprocessor (or, assembler) directives when appropriate, then preprocessed, assembled, and linked into an executable file that is finally programmed to and executed by the relevant device.



7

New Programming "Toolchain" Model

- Now provided on the right is a partial depiction of the generic programming toolchain model we will begin to utilize when programming with the "C" language.
  - **Most everything is as before!** The main difference is that application code will be written primarily with the "C" language, with any "C" code first being converted into an assembly representation that the pre-existing portions of the toolchain can be applied to. A software tool known as a *compiler* does this translation from "C" to assembly code.

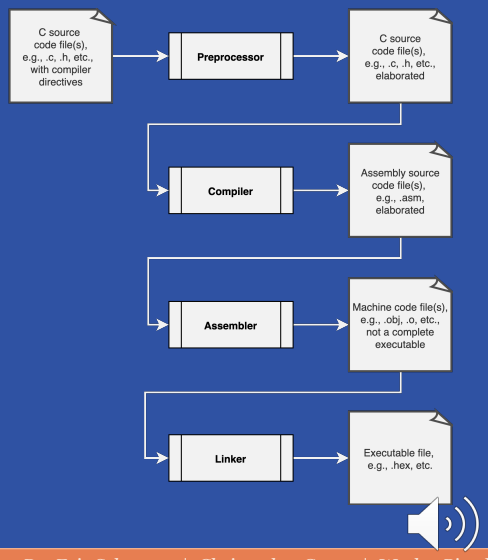


7

8

New Programming "Toolchain" Model, cont.

- With application code being primarily written in "C", it is logical that preprocessor directives be applied within the context of "C" code. Because of this, just as before, a preprocessor will be the first tool executed within the programming toolchain.

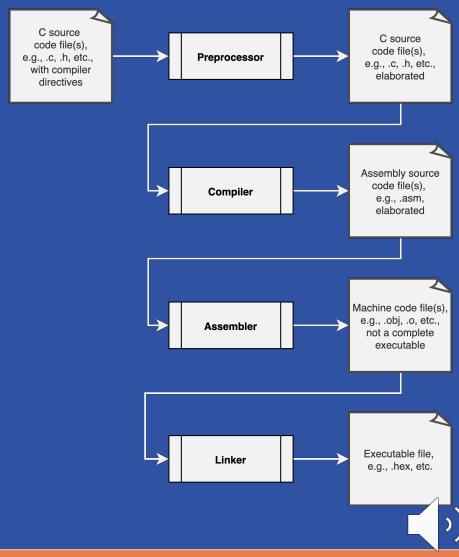


8

9

New Programming "Toolchain" Model, cont.

- Source code written with the "C" language is to exist almost exclusively in files that have the extension ".c".

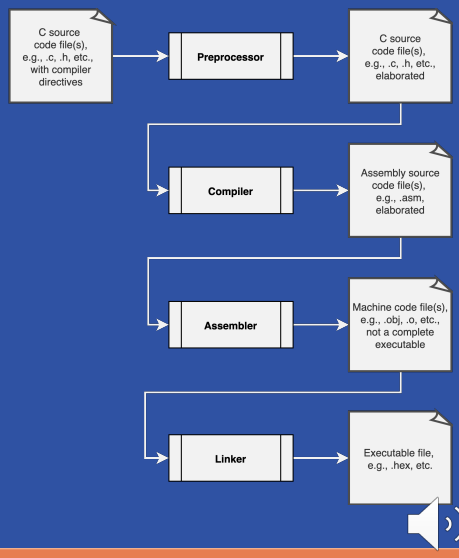


9

10

New Programming "Toolchain" Model, cont.

- When it is desired that a modular structure be created for some program, it is generally helpful to employ *header files*, which are saved with the ".h" file extension.
- In general, structuring "C" code in a meaningful way is an art, not to be taken lightly.



10

11

Some Helpful Resources for Learning "C"

- There are many, many resources out there for learning how to utilize the "C" programming language.
- The primary ones to follow for completeness and accuracy are:
  - "*The C Programming Language (2nd Edition)*", by Brian Kernighan and Dennis Ritchie, the second of whom is the creator of the "C" programming language.
  - The international standard for the "C" programming language, "*ISO/IEC 9899*", which specifies the form and establishes the interpretation of programs written in the "C" programming language. (As of now, we will likely refer to the *1999* version of this standard, although there exist other newer versions.)

11

12

Conclusion

In this lecture, we have covered:

- Motivation for a programming language other than an assembly or machine language.
- A brief introduction to the "C" programming language.
- A listing of some useful resources for learning how to apply the "C" programming language.

NOTE: This lecture has not been a tutorial on how to utilize the "C" programming language; such instruction should be sought throughout other content.

12