FCPA 2022  
  
The C Preprocessor

Student Workbook 10

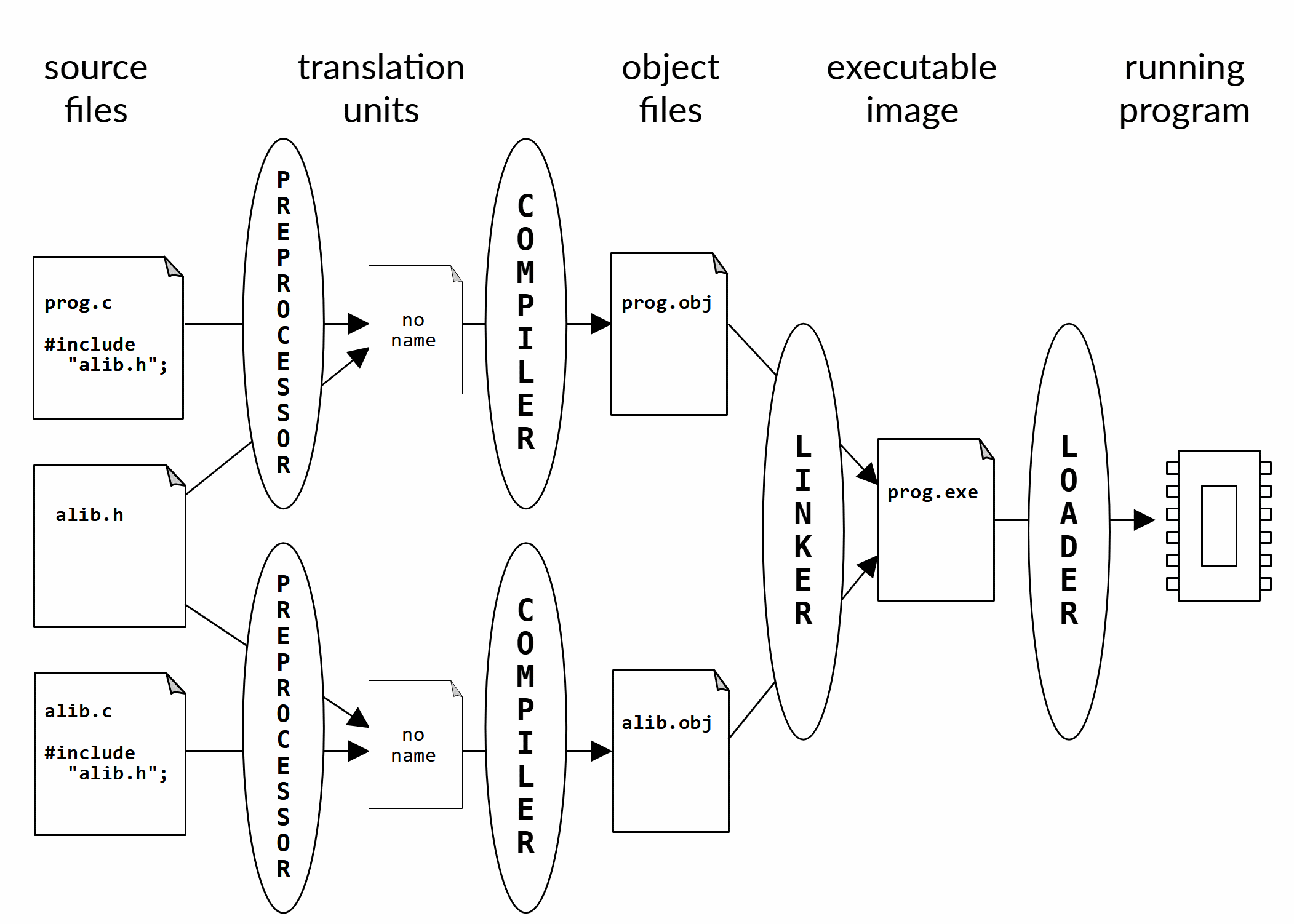
January 21, 2022

Author

Paul Kimball  
Interface Associates

1. Header Files and   
   The Preprocessor

# Separate Compilation



# The C/C++ Preprocessor

## The preprocessor prepares source code for the compiler

### Merges multiple source files into a single stream

### Performs lexical substitutions, called "macros"

### Conditionally defines types, variables and functions

### Produces a single (possibly very large) "compilation unit"

## Is controlled by directives that start with the hash character

### #include

### #define

### #if / #else / #elif / #endif

### #ifdef, #ifndef

### #pragma

### #error

# C Typical Preprocessor Uses

## Including header files

* + And defining header file guards

## Type substitution

## Check and set compile-time constants

## Conditional inclusion of libraries

## Conditional compilation

## Macros

* + Object-like macros
  + Function-like macros

# Preprocessor Directives

## Directive starts with '#' followed by directive name

### Space between # and name is OK

### Directive names are fixed; cannot be defined by programmer

## Each directive appears on its own line

### May be indented for clarity

## Directive and its arguments are terminated with a newline

### A backslash '\' at the end of a line escapes the newline and continues the directive on the next line

### Be very careful ! No spaces after the backslash (!)

# #include

## Incorporates the contents of another file as though wrote it there

* + By convention, called "header files"

### Use ".h" as a file extension

## A header file declares data types, functions, macros and static content provided by a library

* + The merged content must be recognizable as C code
* Filename in angle brackets

#include <theater.h>

* + File is in a system-administered directory, which is known to the compiler - e.g., /usr/lib on Linux
  + Use this syntax for including standard library headers

## Filename in quotes

#include "MovieTicket.h"

* + File is relative to the "include path", starting with the current directory
  + You can specify other directories on the compiler command line
  + Use this syntax for including your own project headers

## Header files can include other headers, recursively

C Standard Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Purpose of library** | **Since** | **Required in Freestanding** |
| <assert.h> | assert and static\_assert (C11) macros | C89 |  |
| <complex.h> | Complex and imaginary numbers |  |  |
| <ctype.h> | Character handling | C89 |  |
| <errno.h> | Errors | C89 |  |
| <fenv.h> | Floating-point environment |  |  |
| <float.h> | Characteristics of floating types | C89 | yes |
| <inttypes.h> | Format conversion of integer types |  |  |
| <iso646.h> | Alternative spellings of logical and bitwise ops |  | yes |
| **<limits.h>** | **Sizes and bounds of integer types** | C89 | yes |
| <locale.h> | Localization | C89 |  |
| **<math.h>** | **Basic mathematics** | C89 |  |
| <setjmp.h> | Nonlocal jumps | C89 |  |
| <signal.h> | Signal handling | C89 |  |
| <stdalign.h> | Alignment |  | yes |
| <stdarg.h> | Variable-length argument lists | C89 | yes |
| <stdatomic.h> | Atomics |  |  |
| **<stdbool.h>** | **Boolean type and values** | C99 | yes |
| <stddef.h> | Common definitions | C89 | yes |
| **<stdint.h>** | **Modern standard integer types** | C99 | yes |
| **<stdio.h>** | **File and console input/output** | C89 |  |
| **<stdlib.h>** | **General utilities** | C89 |  |
| <stdnoreturn.h> | noreturn |  | yes |
| **<string.h>** | **NULL-terminated strings - copy, compare** | C89 |  |
| <tgmath.h> | Type-generic mathematics |  |  |
| <threads.h> | Threads | C11 |  |
| **<time.h>** | **Date and Time** | C89 |  |
| <uchar.h> | Unicode utilities |  |  |
| <wchar.h> | Multibyte/Wide character utilities |  |  |
| <wctype.h> | Wide character classification and mapping |  |  |

* May be included in any order
* May be included more than once
* Described in Chapter 7 and Annex B of Specification
* Let's explore some of these!

Type Substitution

* Used frequently

#include <stdbool.h>

bool done = false; //  \_Bool done = 0;