C Program Compilation

Cn6ion

C Program Compilation

ÅThus far we have kept our programs to a single file

ÅBut we know C programs are made up of multiple files, as we are already including libraries like stdlib.h, etc.

ÅOur latest programs are starting to get pretty big!

ÅThe linked list and binary search tree code examples could really be a library for each data structure...

ÅWhy do we split our programs across multiple files?

Software architecture

Å

Manager
Receptionist
Inventory
+Location: string
+Location: string
+BookRoom()
+ManageStaff()
+GenerateBill()
Guest
Chef
Room
+Id: int
+Name: string
+RoomNo: int
+RoomNo: int

Component-based architecture

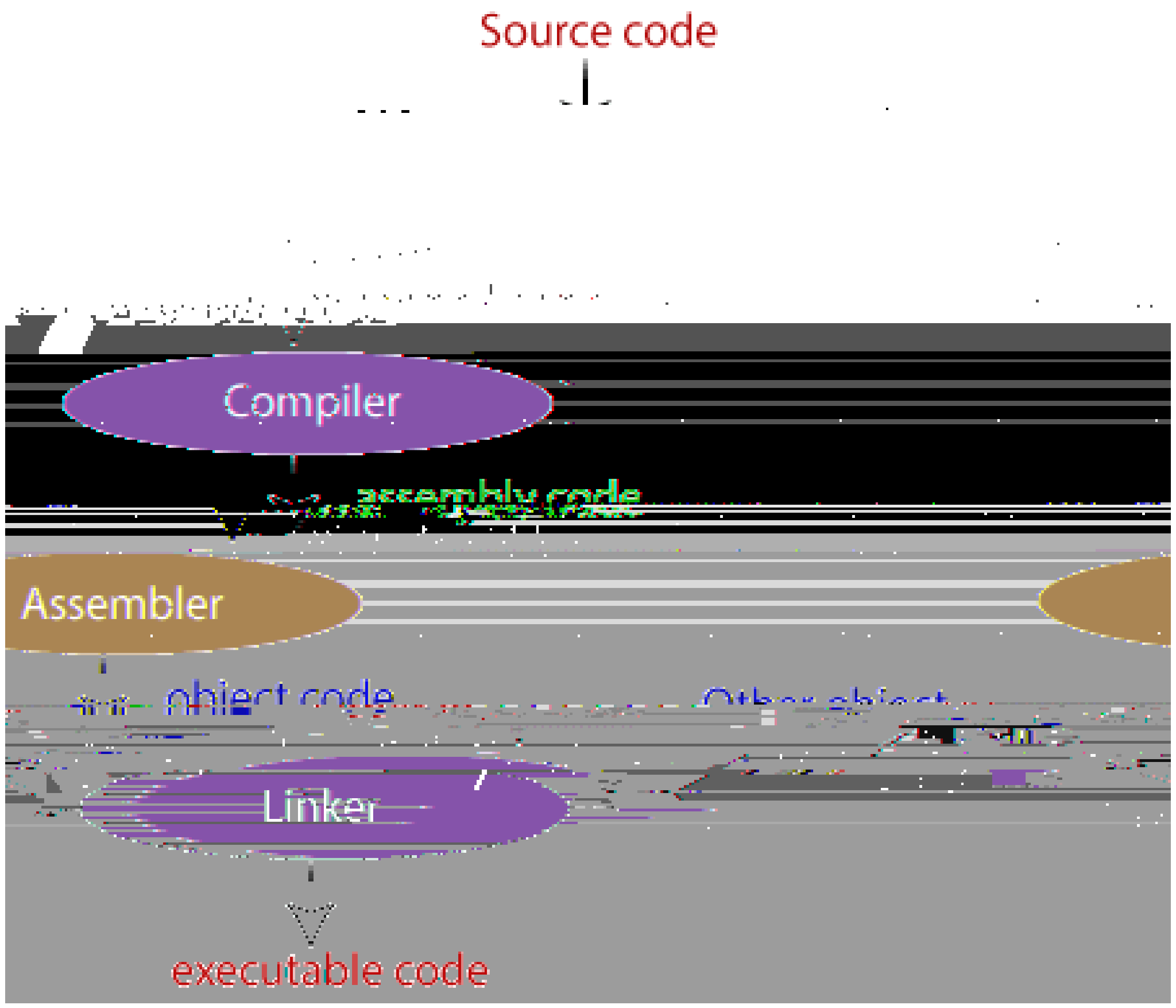
ÅComponents can be re-used in different projects

ÅIf only some components are changed, only those components need to be re-compiled

ÅComponents can be tested individually, which lead

C Compilation

Å



Source: https://static.javatpoint.com/cpages/images/compilation-process-in-c2.png

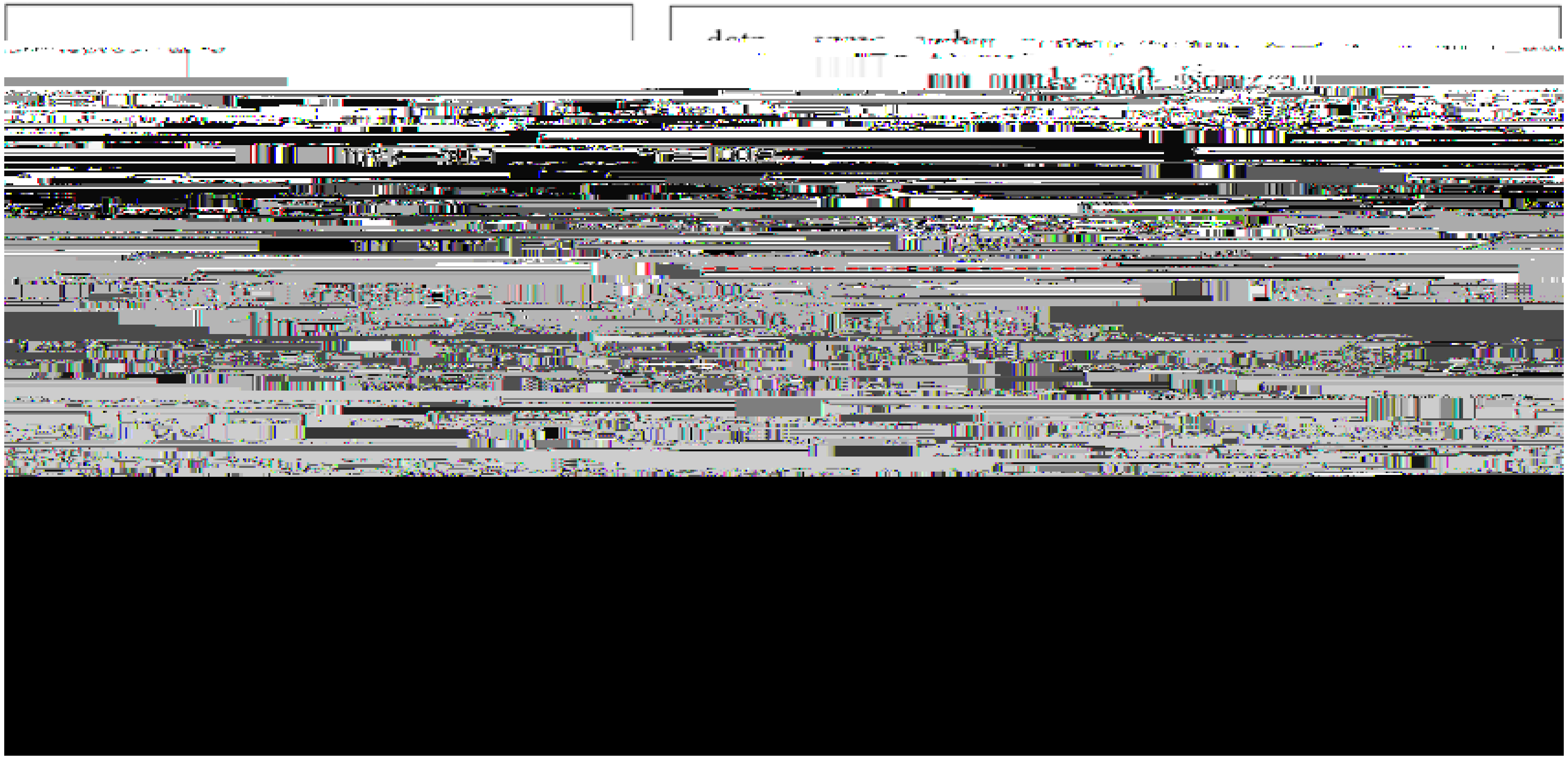
Preprocessor

Å

Compilation stage

Å

C code to assembly code example



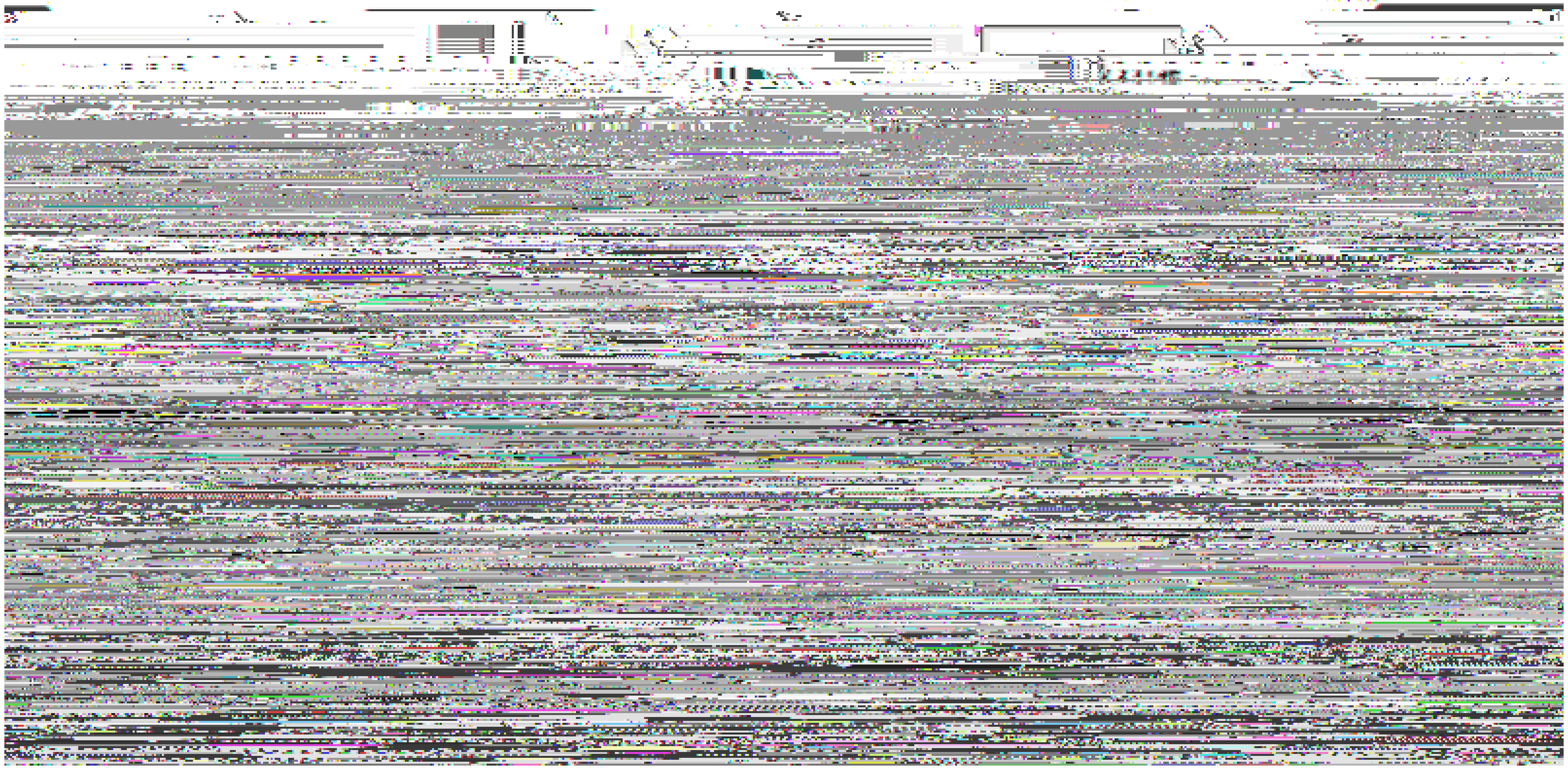
Source: https://www.researchgate.net/profile/Edwin-Sha/publication/220305740/figure/fig2/AS:670708619284480@1536920804237/A-dot-product-C-code-and-its-corresponding-assembly-code-from-TI-C6000-25.png

Assembler

- ÅThe assembler takes as input the assembly code and produces as output machine code (also known as object code)

and pingber

  - ÅUnlike compilation, nothing too complicated is happening here in terms of algorithms formsh 73BT/F2 24.025



Source: https://cdn.educba.com/academy/wp-content/uploads/2019/10/Assembly-Language-vs-Machine-Language-1.png

Linker

ÅThe linker takes as input the various object files required for the program and produces an executable file as output

ÅObject files could include the object files for standard C libraries (e.g. stdio, stdlib, etc.)

- Å These have already been compiled to save time, they just need to be linked at this stage

ÅObject files could also include object files for libraries we have defined ourselves

- Å We need to make sure the compiler knows about these files in order to include them

Source: https://www.tenouk.com/Bufferoverflowc/Bufferoverflow1c.html

Executable file

ÅThe ex(C3i837> BDC q.0061072loA/MCID 3>> BDC q0.001(91(

Compilation and execution

Source: http://fresh2refresh.com/wp-content/uploads/object%20and%20exe.png

Creating C libraries

ÅIn order to create our own C libraries and include them in our program, we need to:

- ÅCreate a .h file that specifies any function declarations, constant values, etc.
- ÅCreate a .c file that contains the function definitions for each function declaration
- ÅInclude the .h file in our .c file containing the main function
- ÅCompile both the .c file containing our main function AND the .c file containing the function definitions for our library

Creating C libraries

Let's create a C library!

main.c

main.c

Notice the #include in main.c!

ÅWe used #include "add.h" instead of #include <add.h>

ÅThe #include <something.h> syntax is used for system library headers

ÅThe #include "something.h" syntax is used for our own libraries created for our program

How do we compile this now?

Å We can run the command:

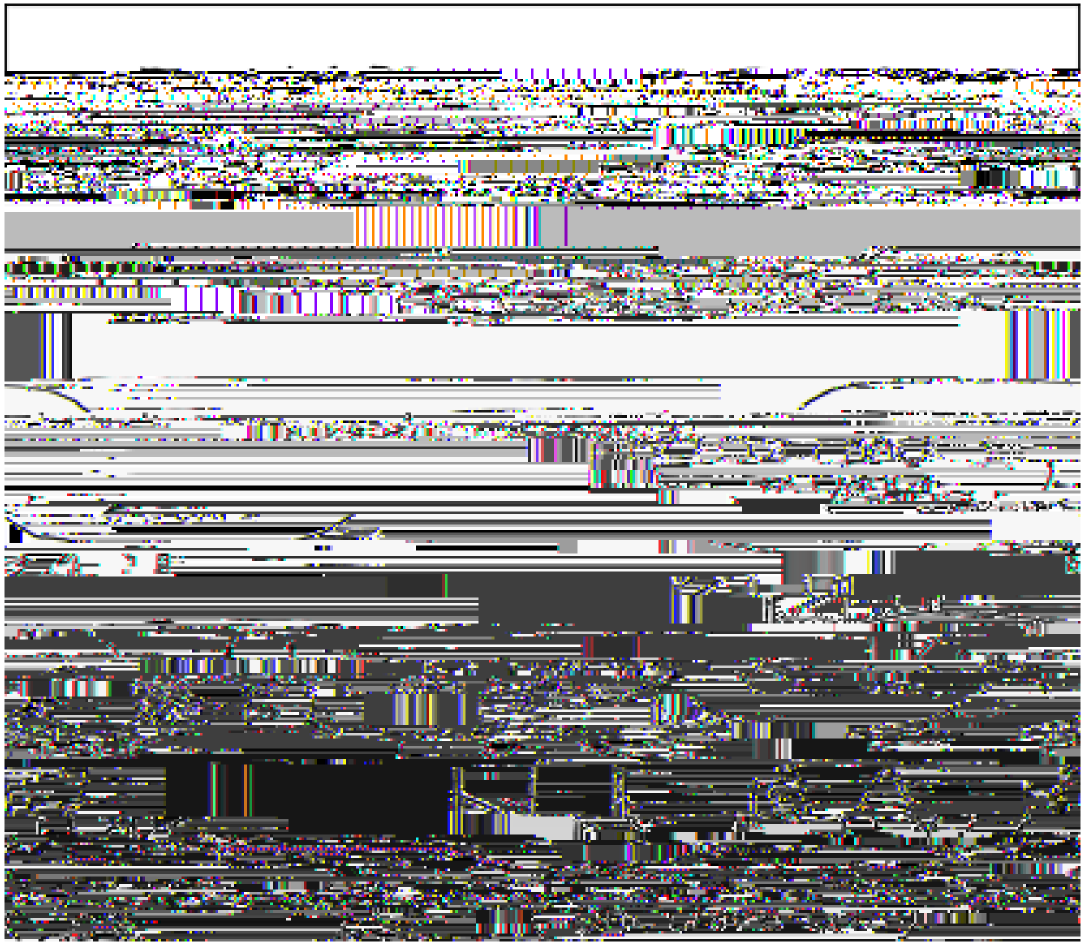
 Å gcc ς

 exe40 r40u 54 te471546neb 54 le

 Å

Compilatioo[illegible]n on [illegible]lPia[illegible]

This famous xkcd comic really does have some truth to it...



Source: https://xkcd.com/303/

Compilation

ÅWe can produce object files with ςc in gcc

Å

Linking

ÅWe can then link these files together with gcc to produce an exectuable

ÅExample:

  Ågcc ςo main main.o add.o

ÅThis will produce an executable main that we can then run

Compilation

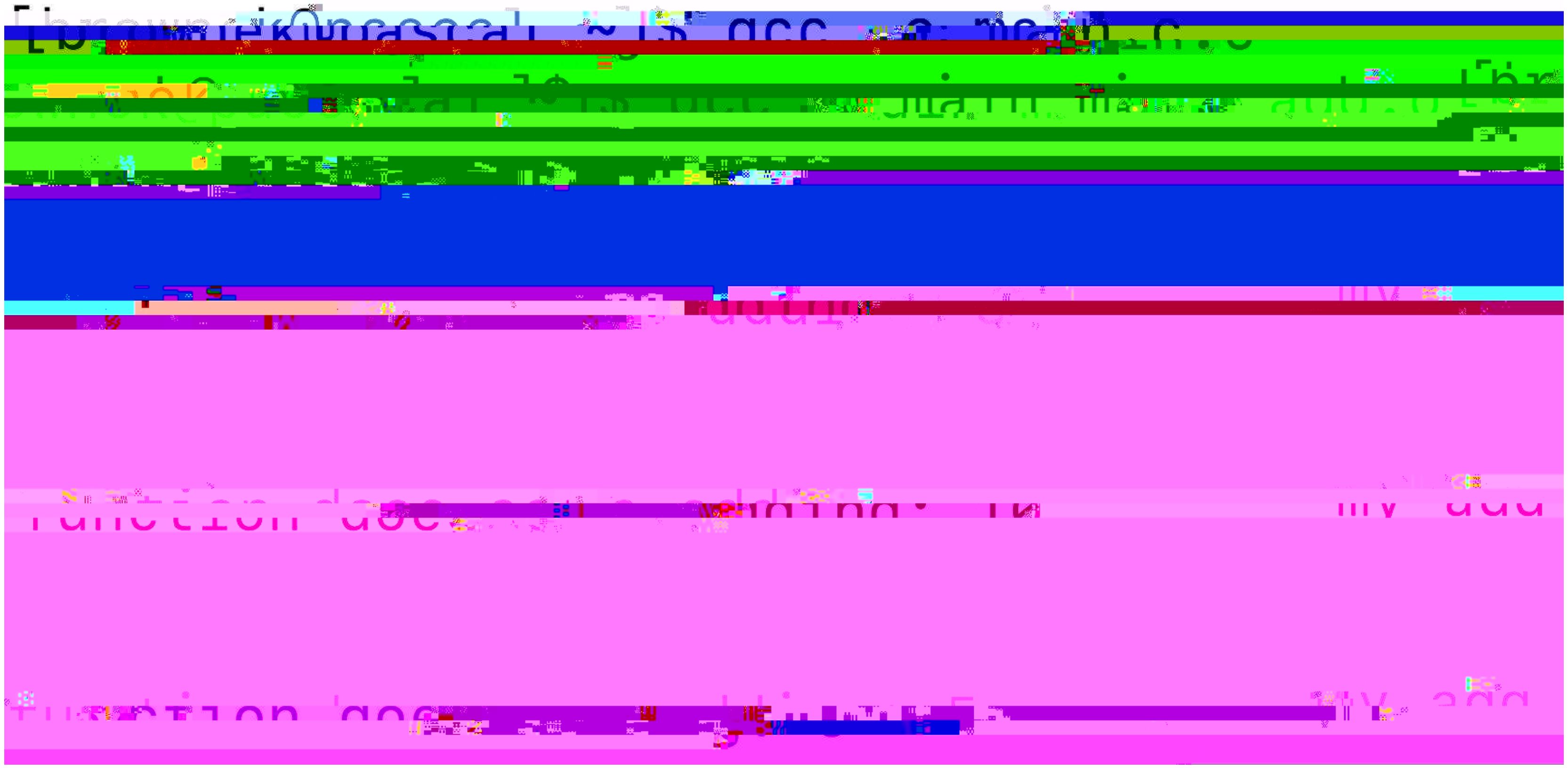
ÅAfter we modify main, there is no need to re-compile add.c

  ÅWe can recompile main.c, and then link the object files again

ÅExample:

  Ågcc ς

Compilation and linking example



```
[brownjek@pascal ~]$ gcc -c main.c
[illegible]ek@pascal ~]$ gcc [illegible] main [illegible] add.o [br[illegible]
[illegible]
[illegible] adding [illegible] my
[illegible]
function does [illegible] adding: [illegible] my add
[illegible]
function does [illegible] my add
```

The same is true if we make modifications to add.c...

We only need to re-compile add.c and then perform another link..

Compilation and linking example

make and make files

Example makefile

[illegible]

Makefiles

ÅThis example makefile:

- ÅExplains which compiler to use (gcc)
- ÅWhat to build (main) and how to build it (using main.o and add.o)
- ÅExplains how to clean up the result of a build (removing main, main.o, add.o)

ÅWe'll talk more about makefiles tomorrow!