GNU FORTRAN

Varun Gumma – 2017A7PS0165H

Muzaffar Ahmed – 2017A7PS0248H

SNS Maneesh Sarma – 2017A7PS0238H

Pavan Srihari Darbha – 2017A7PS0011H

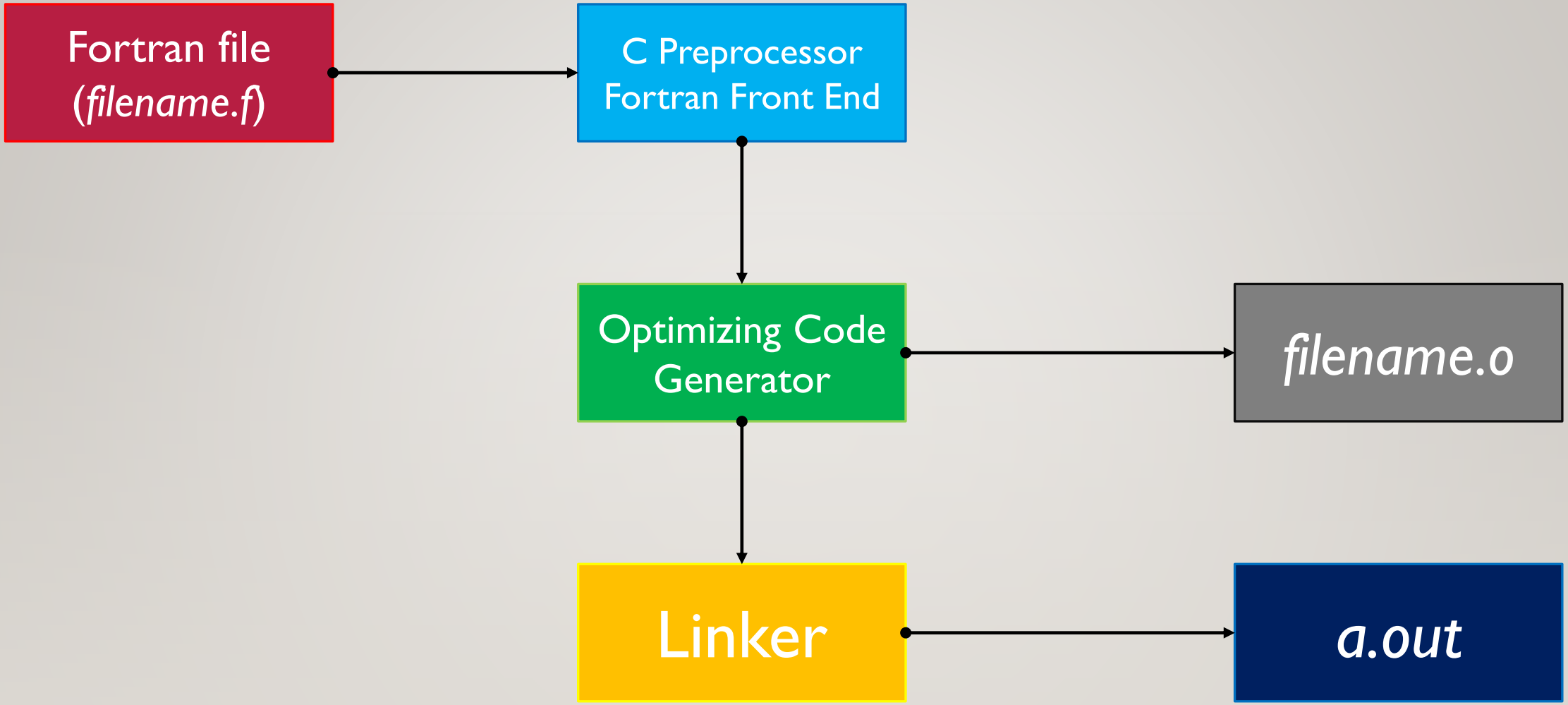
FORTRAN (from Formula Translation) is a general-purpose, compiled, imperative programming language that is best suited to numerical and scientific computing. As such, it has been in use for six decades.

GNU Fortran was developed:

- by the developers at GNU Project, who develop and publish a variety of free software to give users freedom and control in the use of their computers
- between 2001 and 2005, with the initial release in April 2005 replacing the old g77 compiler
- since the principal author and maintainer of g77 stopped work on g77 in 2001, the GFortran project aimed to develop new Fortran front-end and run-time libraries for GCC (GNU Compiler Collection)

GFortran includes full support for Fortran 95, with partial support for Fortran 2003 and 2008.

Fortran was the world’s first High-Level Language, so it is unsurprising.



The Compilation Process

REPRESENTATIONS

- The source file ends with the extensions .f, .F, .for, .FOR or .i
- The compiler has an integrated C preprocessor that provides full cpp capabilities.
- The compiler produces a linkable object file when the –c command is specified. It takes the original filename but the output has the extension .o
- The executable file is a.out by default.
- Another name can be specified for the executable using the –o *name* command.

STRENGTHS

- ✓ The efficiency of compiled code is quite high because Fortran is straight-forward to compile and techniques for handling it have reached a great degree of refinement.
- ✓ It’s free, so easily available to all.
- ✓ Being Open-Source, the development is fast since anyone can contribute.
- ✓ GNU Fortran can be used in programs with multiple threads, while guaranteeing thread safety.
- ✓ The compiler does not evaluate all parts of an expression, if the result can be arrived at without needing all the parts.
- ✓ Asynchronous I/O is supported if the program is linked against the POSIX thread library. If not, all I/O performed is synchronous.
- ✓ For consistency, GNU Fortran uses buffered I/O in order to improve performance. This buffer is flushed automatically when full and whenever necessary.

WEAKNESSES

- GNU Fortran does not support the latest Fortran standards (Fortran 2018). In fact, it does not even offer full support to Fortran 2003 and 2008 standards.
- When it comes to durability, GNU Fortran does not ensure that data is committed to stable storage. It is entirely left to the programmer to code for it if necessary.
- GNU Fortran creates a name.mod for each module name. This confuses the GNU make into thinking they are Modula2 source files. The user must manually over-ride this with a command.
- Block Data causes trouble with linking and initializing under GNU Fortran. Sharing common blocks across Fortran tasks is somewhat tedious.
- Some programs make use of overflow of variables for a desired result. GNU Fortran throws an Arithmetic Overflow error immediately and it must be forced to accept it manually.
- GNU Fortran’s error messages do not comply to patterns in emacs's compilation-error-regexp-alist.

In Conclusion…

Fortran has been around for so long, it is to be expected that it runs on just about any system, give or take. Likewise, GNU Fortran is also compatible with a wide variety of systems. A necessity, given that it was an Open Source project to begin with. As such, given that Fortran is so widespread, so is a reliable compiler for the same.

OTHER COMPILERS

- Embedded systems: Keil, Code Composer Studio, GNU/GCC and Ride 7
- Small devices: SDCC (Small Device C Compiler), TCC (Tiny C Compiler)
- Parallel programming: ROSE, Intel Fortran Compiler, Intel C++ Compiler
- Graphics: Intel Graphics Compiler