

GNU/Linux - Executables

Class 14-16

Prasad H. L.

April 30, May 2, May 7 of 2008

This document is distributed under GNU FDL. To obtain a full copy of GNU FDL [\[\[click here\]\]](#).

Review

- GNU/Linux Boot
- Executable Creation, Load and Execution
- Executable Format - Introduction

Contents

1 Linux Executable	1
2 Make Utility	3
2.1 Introduction	3
2.2 makefile	3
3 Summary	4

1 Linux Executable

ELF Binary

- Executable and Linking Format
- Sections
 - .init, .fini
 - .data, .bss, .rodata
 - .text

- .dynamic, .dynstr, .dynsym

- Loader - ld-linux.so.2
- Syscall points - linux-gate.so.1
- Commands - ldd, nm, objdump, readelf, size, strings.

GNU/Linux Loader

Tasks of a Loader

- load the executable sections to process segments,
- find and load the shared libraries needed by a program,
- prepare the program to run, and
- then run it!

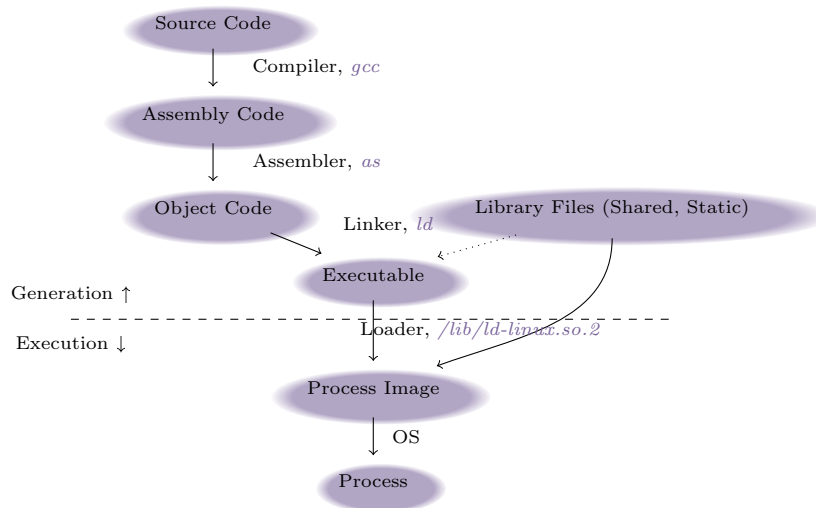
Types

- ld.so
- ld-linux.so

GNU/Linux System Call Point - linux-gate.so.1

- Contains entry point for system calls

Executable Life Cycle



2 Make Utility

2.1 Introduction

What is make

Necessity

- Many stages lead to product
- Inverted tree structure
- Various branches culminate to final product
- Redoing all branches on change at any point is unnecessary
- *make* helps automate the above for minimal compilation time

2.2 makefile

Definition of make tree

Tree Definition

the make tree is define in a file called *Makefile*

- Technically a ‘Directed Graph’¹
- Defined as a *target* and its *prerequisites*
- *Directives* define the way to obtain the node from the dependencies

Makefile structure

Elements of Makefile

- Explicit rule
- Implicit rule
- Variable definition
- Comment

¹without loops

Makefile structure

The structure of Makefile

The basic block of a rule syntax is as follows:

```
target:prerequisites
    command
```

- Note that the target is without a tab
- The target and prerequisites are separated by a :
- The rules are written with an initial tab

Automatic variables

Useful for writing Makefiles

`$$` The file name of the target of the rule
`$$%` The target member name, when an archive member
`$$<` The name of the first prerequisite.
`$$?` The names of prerequisites newer than target
`$$^` The names of all prerequisites.
`$$+` Prerequisites just as listed in the makefile
`$$|` Order-only prerequisites.
`$$*` The stem with which an implicit rule matches

How make knows what to build?

What is built?

- A target is built if the prerequisites have a *timestamp* later than the target.
- If the prerequisites have a timestamp before the target and if prerequisites are themselves targets then they are first built

3 Summary

Summary

- GNU/Linux Executable
- GNU/Linux Loader
- gmake Utility

References

References

- [1] *man ld-linux, man:/ld-linux*
- [2] *info make, info:/make*
- [3] *info binutils, info:/binutils*