Institute of Engineering and Technology, Ahmedabad University

Operating System

Seminar Summary

GNU Automake

Author:
Rahul Patel

Supervisor:
Dr. Hemal SHAH

April 23, 2015

Contents

1 Background

The topic to scrutinize was *GNUAutomake*. In the process of understanding what is *GNUAutomake*, I came across few other terms which are like prerequisite to understand the *Automake* and appreciate the whole package of tools namely *Autotools*. *Autotools* is package which contains other sub-packages like *Automake*, *Autoconf*, *Libtool*, *Gettext*. The name consists of namely two parts auto and tools and it may occur to any layman after reading it out that it must be bringing some automation or trying to reduce manual efforts in the current scenario. The primary objective of this package was to automate the build process making it platform independent.[1][2]

1.1 What is *GNU*?

GNU is free operating system available for use build upon *Unix*.[1] Along with this operating system there are several software packages available which are useful not only for this operating system but also for any other *Unix* based operating system. *GNU Autotools* are one of them which are useful for building software across various platforms without manual intervention.[1]

1.2 *Make* Utility

The*make*utility automatically determines which pieces of a large program need to be recompiled, and issues commands to recompile them. To use*make*, you must write a file called *makefile*,that describes the relationships among files in your program and provides commands for updating each file. The*make*program uses the *makefile* data base and the last-modification times of the files to decide which of the files need to be updated. For each of those files, it issues the recipes recorded in the data base.[3][5]

2 *GNUAutomake*

Before understanding this we must also have a brief background about what *GNUAutoconf*.

2.1 *Autoconf*

Autoconf is a tool for producing shell scripts that automatically configure software source code packages to adapt to many kinds of Posix-like systems. The primary goal is not to make the generation of*configure*automatic for package maintainers; rather, the goal is to make*configure*painless, portable, and predictable for the end user.[1][4][5]

2.2 *Automake*

Automake is a tool for automatically generating*Makefile.ins* from files called*Makefile.am*. *Makefile.am* is a user defined file containing minimum information. The generated*Makefile.ins* are compliant with the *GNUMakefile* standards.*Automake* does constrain a project in certain ways; for instance, it assumes that the project uses *Autoconf*. This *Makefile.am* is further used to generate *makefile*.

2.3 Pros and Cons

As such there are no Pros and Cons. The topic is highly subjective and mixed views are available on the *Internet*. To briefly summarize them, *Autotools* as a whole makes the cross platform building process easy and less cumbersome by making it platform independent. The major drawbacks are they still use the syntax of M4 macro language which is less user friendly , hence it's difficult to debug and code them at times. Also it's difficult to use them to build software on *Windows* platform. One might not want to consider as this as valid point as it was meant for it, but as a user you would like to have a software which can work on major platform easily

3 Current Build Systems

Automake was a part of build system package. Hence there is no such module available which is analogous to *Automake*. Hence I would like to compare this build system with some other build systems, namely *SCons* and *Gradle*.

3.1 SCons

It works on python and hence is simple to code and has auto-configuration abilities. It's a single step solution unlike *Autotools* using *Autoconf*, *Automake*, etc with no intermediate steps like generating a makefile. Run the script and the program is build.[6]

3.2 Gradle

It supports build for around 60 languages. Gradle is highly customizable according to needs of the user. It has good user support; training and workshops are organized frequently. One interesting thing to note is, *Android* recently shifted to *AndroidStudio* which uses *Gradle* at back end to build programs.[7]

References

[1] Official GNU website - `http://www.gnu.org`

[2] Official Automake Manual - `http://www.gnu.org/s/automake/manual/automake.html`

[3] Make Utility - `https://www.gnu.org/software/make/`

[4] Configure Script wiki - `http://en.wikipedia.org/wiki/Configure_script`

[5] GNU build system wiki - `http://en.wikipedia.org/wiki/GNU_build_system`

[6] Official SCons website - `http://www.scons.org`

[7] Official Gradle website - `https://gradle.org`