GNUstep Make Version 2

Nicola Pero, Fosdem 2007

Introduction

This presentation is about GNUstep Make:

- Origins and Design (1997)
- History (1997-2006)
- Version 2 (2007)

Origins of GNUstep

- NEXTSTEP 1.0 (1989)
- OpenStep Spec (1994)
- GNUstep launch (1995)

Origins of GNUstep Make

GNUstep Make (1997) is implemented to:

- Support a **custom filesystem structure**
- Provide a **building system**

The same software is used to do two **different** things.

Custom Filesystem Structure

- NEXTSTEP-like filesystem structure
- NEXTSTEP-like project types (app wrappers, bundles, palettes, etc)
- Support for multiple ObjC runtimes and OpenStep libs on the same system (“library combo”)
- Support for fat binaries
- Support for debug/profile/static libs

GNUstep Filesystem Layout

```
/usr/GNUstep/System/Applications
/usr/GNUstep/System/Tools
/usr/GNUstep/System/Library/Headers
/usr/GNUstep/System/Library/Libraries
/usr/GNUstep/System/Library/Bundles
/usr/GNUstep/System/Library/...

/usr/GNUstep/Local/Applications
/usr/GNUstep/Local/Tools
/usr/GNUstep/Local/Library/Headers
/usr/GNUstep/Local/Library/Libraries
/usr/GNUstep/Local/Library/Bundles
/usr/GNUstep/Local/Library/...

/usr/GNUstep/Network/...

/home/nicola/GNUstep/...
```

Non-Flattened Filesystem

```
/usr/GNUstep/Tools/openapp
/usr/GNUstep/Tools/ix86/linux-gnu/gnu-gnu-gnu/autogsdoc

/usr/GNUstep/System/Library/Libraries/ix86/linux-gnu/gnu-gnu-gnu/libgnustep-base.so.1.14.0

/usr/GNUstep/System/Library/Headers/gnu-gnu-gnu/Foundation/NSData.h

/usr/GNUstep/System/Library/Headers/gnu-gnu-gnu/ix86/linux-gnu/GNUstepBase/GSConfig.h

/usr/GNUstep/System/Library/Bundles/SSL.bundle/ix86/linux-gnu/gnu-gnu-gnu/SSL
/usr/GNUstep/System/Library/Bundles/SSL.bundle/Resources/
```

Building System

- Simplicity of usage: 'make' to compile a project
- Elegant and simple makefiles
- Portability

```
include$(GNUSTEP_MAKEFILES)/common.make

TOOL_NAME = client server
client_OBJC_FILES = client.m common.m
server_OBJC_FILES = server.m common.m

include $(GNUSTEP_MAKEFILES)/tool.make
```

GNUstep Make Initial Design

Two separate (but related) things to implement:

- **Custom filesystem**: NEXTSTEP filesystem/library-combo/fat binary support layer (runtime)
- **Makefile library**: building system (build time)

Custom Filesystem (Problems)

Problems:

- NEXTSTEP-like filesystem: programs can't be found, libraries can't be found
- NEXTSTEP-like bundles: app executables can't be found
- Library combos and fat binaries: correct version of executables/libraries can't be found

Custom Filesystem (Solution)

Layer of shell scripts & wrappers:

- Users must source GNUstep.sh when they login (determines system type/config, sets PATH and linker paths)
- Applications are started using 'openapp TextEdit.app'
- Tools are started using 'opentool defaults'

Custom Filesystem (Pros)

Advantages:

- Nice NEXTSTEP-like filesystem
- Fat binary support, can switch library combo on the fly
- Can mount GNUstep on different machines from single fat, shared network filesystem!
- Can easily keep multiple GNUstep installations

Custom Filesystem (Cons)

Disadvantages:

- It can not be used without a lot of shell scripts
- It installs everything in non-standard locations

It's a “NEXTSTEP emulator” running on your system.

Makefile Library

- Everything done at runtime
- Depends on GNUstep.sh being sourced
- Support for lots of systems
- Multiple recursive invocations of the same makefile

History (1997-2001a)

- 0.6.0 (1999)
- 0.6.6 (2000) Added System root directory
- 0.9.1 (2001) Flattened directory structure available
- 0.9.2 (2001) System, Local, Network root directories can be relocated

History (2001b - 2002)

- 1.0.0 (2001): Optimizations
- 1.0.1 (2001): Simplified library-combo (was gnu-gnu-gnu-gnu, becomes gnu-gnu-gnu)
- 1.2.0 (2001): Rewritten core recursive make invocations
- 1.3.0 (2002): Separated Master & Instance invocation code

History (2003-2005)

- 1.7.0 (2003): GNUstep Filesystem Hierarchy
- 1.7.2 (2003): Flattened becomes the default
- 1.10.0 (2004): User root configuration via GNUsteprc
- 1.11.0 (2005): All root directories configured via GNUstep.conf file
- 1.11.2 (2005): GNUstep.sh no longer needed

History (2006-2007)

- 1.13.0 (2006): debug libraries have the same name as standard libraries
- 2.0.0 (2007): will be released soon, currently trunk is 1.98.0 (pre-2.0.0 release).

2.0.0 (2007) [1]

Support for parallel debug/profile builds was dropped:

- All apps have the same name (always Gorm.app)
- All build dirs have the same name (always ./obj)
- No more C tools

2.0.0 (2007) [2]

Configurable filesystem layout:

- Arbitrary filesystem layout can be used
- System/Local/Network/User root directories are replaced with System/Local/Network/User domains
- GNUSTEP_INSTALLATION_DOMAIN must be used instead of GNUSTEP_INSTALLATION_DIR

2.0.0 (2007) [3]

Other enhancements such as:

- Support for 'make DESTDIR=xxx'
- Admin apps/tools directories added
- gnustep-config script (easier integration with other building systems)
- Precompiled header support

Using GNUstep Make v2

Supports a variety of options:

- NEXTSTEP-like filesystem layout with fat binaries (requires GNUstep.sh and shell wrappers)
- NEXTSTEP-like filesystem layout with no fat binaries (requires PATH, linker path)
- Native filesystem layout (no requirements)

Setup: Flattened Case

- Add System Tools and Local Tools to PATH
- Add System Libraries and Local Libraries to linker path (eg, /etc/ld.so.conf)

If the filesystem layout is **native**, it's all already done.

Setup: Non-Flattened Case

You need to source GNUstep.sh before using the system --

```
. /usr/GNUstep/System/Library/Makefiles/GNUstep.sh
```

Setup: the User Domain

The user domain is special:

- It's a subdirectory of your home directory, so:
- User Tools is not normally in PATH
- User Libraries is not normally in the linker path

So it requires GNUstep.sh to work.

Setup: Multiple Installations

- GNUstep Make and GNUstep Base use a central config file to determine the filesystem layout.
- Normally it is /etc/GNUstep/GNUstep.conf
- Other files can be used by setting GNUSTEP_CONFIG_FILE

Setup: Multiple Installations(2)

- Use `./configure -with-config-file=xxx` option to specify the config file to use
- Use GNUSTEP_CONFIG_FILE to have everything use that installation at runtime:

```
export GNUSTEP_CONFIG_FILE=/etc/MyOtherGNUstep.conf
```

Thank you

For more information

GNUstep web site (http://www.gnustep.org)