Developing Platform Independent Software using the AutoTool Suite

Jason But

Outline

- Why develop Platform Independent code
- From the users perspective
- From the developers perspective
- The Autotools Suite
  - Automake
  - Autoconf
- The NAM (**N**ot **A**uto**M**ake) System
  - Why
  - How to use it

Why Platform Independent Code?

- Most software we develop is likely to be tools to gather or analyse data
  - Smaller
  - Possibly only used by you
- It may be useful enough to release
  - E.g. pckhisto and netsniff
- If others use FreeBSD
  - Environment could be configured differently, your Makefile or script might not work
- Other Platform OSs
  - Linux
  - MacOS X
  - Windows – God forbid

SWINBURNE UNIVERSITY OF TECHNOLOGY

Why Platform Independent Code?

- Developing Platform Independent Software
  - The source code
    - Generic
    - Use platform independent libraries
    - Write standards compliant code (for different compilets)
    - Where necessary, group platform dependent code into a small file set so different versions can be compiled in
  - The build environment
    - Locations of tools and libraries
    - Names of tools and libraries
    - Version of make
    - How to install

SWINBURNE UNIVERSITY OF TECHNOLOGY

Why Platform Independent Code?

- Need to distribute a project that
  - Examines the build system and determines if it can build the software
  - Creates a standard means of building and installing the software
    - Typically make, but make differs across platforms
  - Source code written to be able to compile on different platforms and with different compilers
- Many options but fast becoming standard is the:
  - *configure*/*make*/*make* install cycle

The Users Perspective

- Simplified Download/Compile/Installation Cycle
  - Download and uncompress source code
  - Execute

```
./configure
su root
make
make install
```

- Consistent across all platforms and increasingly among distributed software

The Users Perspective

- The **configure** script scans the system and build a (set of) **Makefile(s)** specific to the platform under consideration
- The standard **Make** tool is then used to compile and install the software
- Advantages
  - Platform specific instructions are automatically handled
  - User doesn't have to worry about changing compile or install options
  - **configure** can check for required libraries/software/features and fail with an appropriate error message

The Developers Perspective

- Simplified Support Scenario
  - The same package is compatible with a number of different systems
- Complex Development
  - How to write the **configure** script
  - What to check for and how
  - How to generate platform independent **Makefiles**
  - Maintenance of installation system

The Developers Perspective

- **configure** must run on all systems – have to use a standard scripting language (/bin/sh)
- Different systems:
  - ☐ Have libraries and tools installed in different locations
  - ☐ Install your application to different locations
  - ☐ Have different Make systems – BSD Make is substantially different to GNU Make

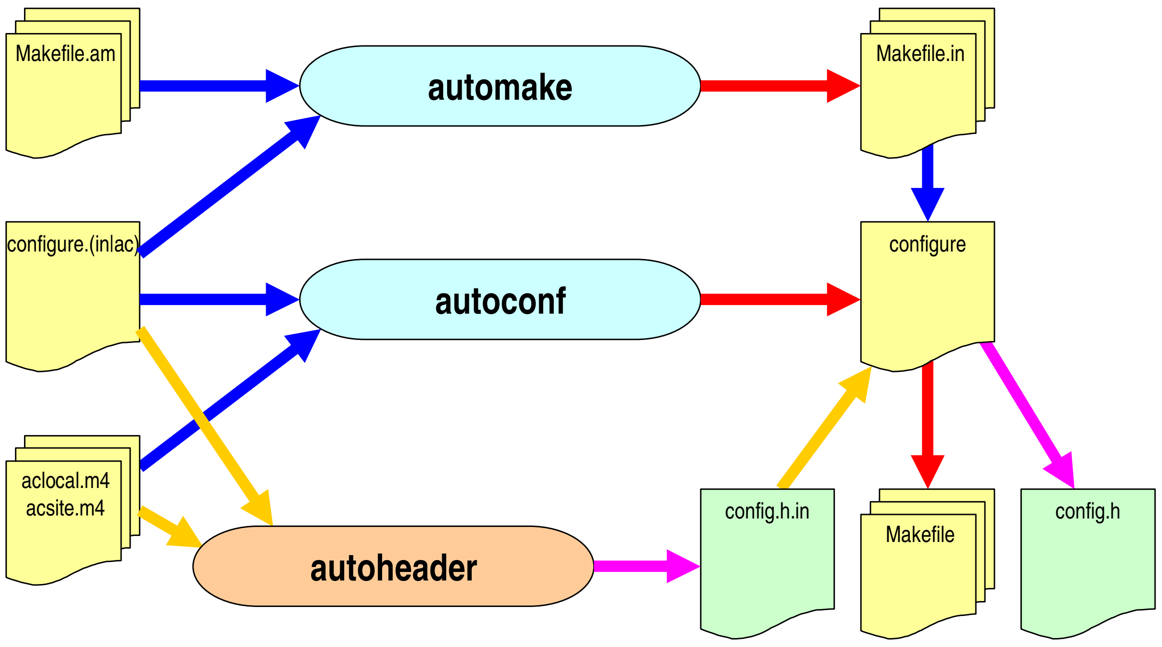
SWINBURNE UNIVERSITY OF TECHNOLOGY

The Autotools Suite

- The GNU Autotools suite was developed to help simplify the task of distribution of platform independent code
  - ☐ **automake**
  - ☐ **autoconf**
  - ☐ **autoheader**
  - ☐ **libtool**
- Of most interest is **automake** and **autoconf**, used to generate a system independent **configure** script which can subsequently be used to generate system independent **Makefile(s)**

SWINBURNE UNIVERSITY OF TECHNOLOGY

Autotools – Overview



The Autotools Suite

- The concept behind the GNU Autotools suite is a good one:

  Remove the task of creating system checks and developing system independent build environments from the developer through the use of a simple tool set.

- But just how good are the individual tools within the suite – particularly **automake** and **autoconf**

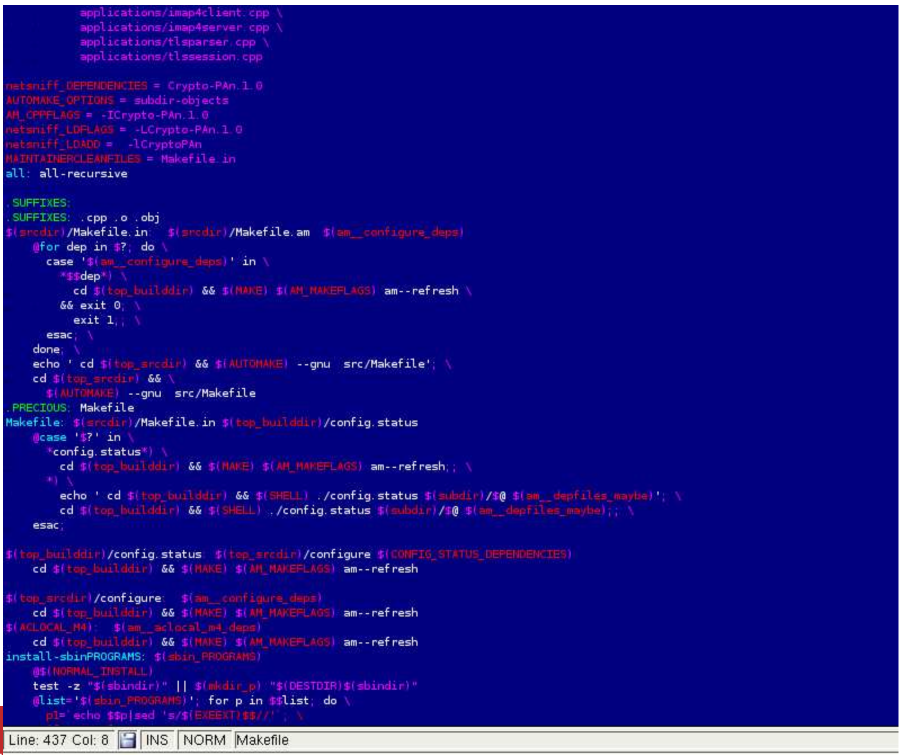
Automake

- Input files
  - *Makefile.am*
    - List of executables and libraries to build and sources for each target
  - *configure.(ac|in)*
    - List of *Makefile.am* files to consider
    - List of macros to implement
    - Determination of which rules to include
- Output files
  - *Makefile.in*
    - Input for **configure** script

Automake

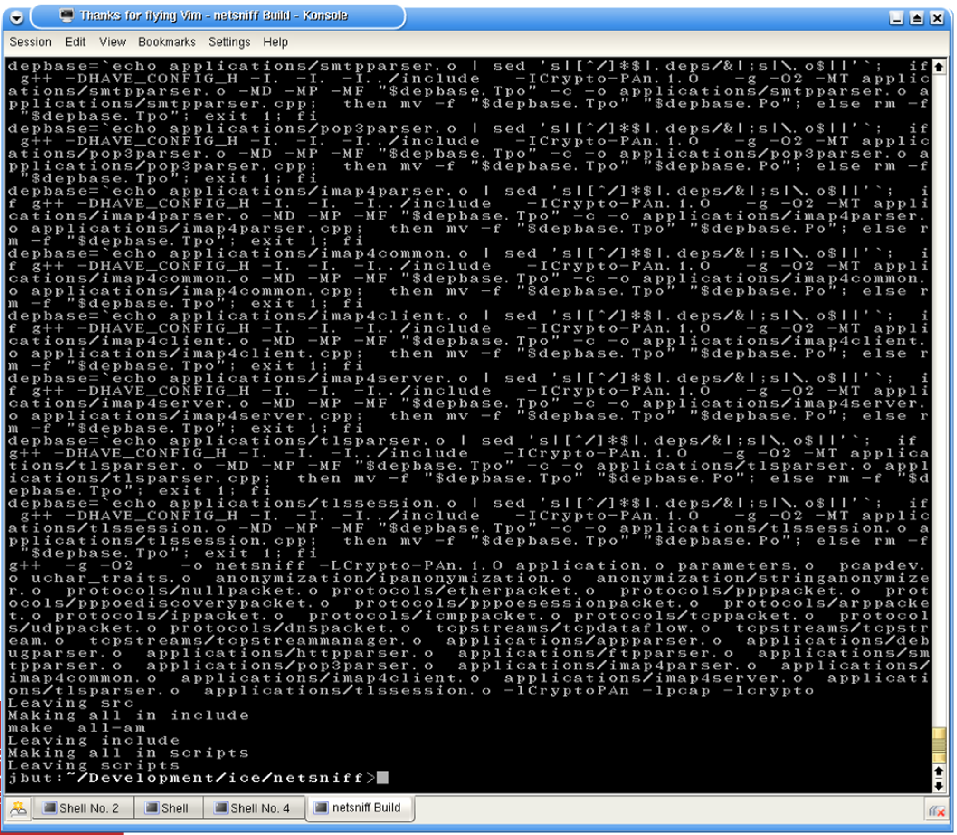
- Advantages
  - *Makefile.am* format is simple and easy to read
  - Don't have to worry about writing a *Makefile*
  - Supports a large number of different types of targets
- Disadvantages
  - Difficult to add extra rules – as discovered when Kris tried to add support for pre-compiled headers with **gcc3.4**
  - Generated *Makefile.in* files are complex and difficult to follow
  - Final *Makefile(s)* difficult to read
    - Debugging problems
    - Understanding build process
  - Running **make** produces ugly output

Automake – *Makefile* sample



```
          applications/imap4client.cpp \
          applications/imap4server.cpp \
          applications/tlsparser.cpp \
          applications/tlssession.cpp

netsniff_DEPENDENCIES = Crypto-PAn.1.0
AUTOMAKE_OPTIONS = subdir-objects
AM_CPPFLAGS = -ICrypto-PAn.1.0
netsniff_LDFLAGS = -LCrypto-PAn.1.0
netsniff_LDADD =  -lCryptoPAn
MAINTAINERCLEANFILES = Makefile.in
all: all-recursive

.SUFFIXES:
.SUFFIXES: .cpp .o .obj
$(srcdir)/Makefile.in:  $(srcdir)/Makefile.am  $(am__configure_deps)
    @for dep in $?; do \
      case '$(am__configure_deps)' in \
        *$$dep*) \
          cd $(top_builddir) && $(MAKE) $(AM_MAKEFLAGS) am--refresh \
        && exit 0; \
          exit 1;; \
      esac; \
    done; \
    echo ' cd $(top_srcdir) && $(AUTOMAKE) --gnu  src/Makefile'; \
    cd $(top_srcdir) && \
      $(AUTOMAKE) --gnu  src/Makefile
.PRECIOUS: Makefile
Makefile: $(srcdir)/Makefile.in $(top_builddir)/config.status
    @case '$?' in \
      *config.status*) \
        cd $(top_builddir) && $(MAKE) $(AM_MAKEFLAGS) am--refresh;; \
      *) \
        echo ' cd $(top_builddir) && $(SHELL) ./config.status $(subdir)/$@ $(am__depfiles_maybe)'; \
        cd $(top_builddir) && $(SHELL) ./config.status $(subdir)/$@ $(am__depfiles_maybe);; \
    esac;

$(top_builddir)/config.status: $(top_srcdir)/configure $(CONFIG_STATUS_DEPENDENCIES)
    cd $(top_builddir) && $(MAKE) $(AM_MAKEFLAGS) am--refresh

$(top_srcdir)/configure:  $(am__configure_deps)
    cd $(top_builddir) && $(MAKE) $(AM_MAKEFLAGS) am--refresh
$(ACLOCAL_M4):  $(am__aclocal_m4_deps)
    cd $(top_builddir) && $(MAKE) $(AM_MAKEFLAGS) am--refresh
install-sbinPROGRAMS: $(sbin_PROGRAMS)
    @$(NORMAL_INSTALL)
    test -z "$(sbindir)" || $(mkdir_p) "$(DESTDIR)$(sbindir)"
    @list='$(sbin_PROGRAMS)'; for p in $$list; do \
      p1=`echo $$p|sed 's/$(EXEEXT)$$//'`; \
```

Automake – make output sample



```
depbase=`echo applications/smtpparser.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`;  if
 g++ -DHAVE_CONFIG_H -I. -I. -I../include  -ICrypto-PAn.1.0  -g -O2 -MT applic
ations/smtpparser.o -MD -MP -MF "$depbase.Tpo" -c -o applications/smtpparser.o a
pplications/smtpparser.cpp;  then mv -f "$depbase.Tpo" "$depbase.Po"; else rm -f
 "$depbase.Tpo"; exit 1; fi
depbase=`echo applications/pop3parser.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`;  if
 g++ -DHAVE_CONFIG_H -I. -I. -I../include  -ICrypto-PAn.1.0  -g -O2 -MT applic
ations/pop3parser.o -MD -MP -MF "$depbase.Tpo" -c -o applications/pop3parser.o a
pplications/pop3parser.cpp;  then mv -f "$depbase.Tpo" "$depbase.Po"; else rm -f
 "$depbase.Tpo"; exit 1; fi
depbase=`echo applications/imap4parser.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`;  i
f g++ -DHAVE_CONFIG_H -I. -I. -I../include  -ICrypto-PAn.1.0  -g -O2 -MT appli
cations/imap4parser.o -MD -MP -MF "$depbase.Tpo" -c -o applications/imap4parser.
o applications/imap4parser.cpp;  then mv -f "$depbase.Tpo" "$depbase.Po"; else r
m -f "$depbase.Tpo"; exit 1; fi
depbase=`echo applications/imap4common.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`;  i
f g++ -DHAVE_CONFIG_H -I. -I. -I../include  -ICrypto-PAn.1.0  -g -O2 -MT appli
cations/imap4common.o -MD -MP -MF "$depbase.Tpo" -c -o applications/imap4common.
o applications/imap4common.cpp;  then mv -f "$depbase.Tpo" "$depbase.Po"; else r
m -f "$depbase.Tpo"; exit 1; fi
depbase=`echo applications/imap4client.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`;  i
f g++ -DHAVE_CONFIG_H -I. -I. -I../include  -ICrypto-PAn.1.0  -g -O2 -MT appli
cations/imap4client.o -MD -MP -MF "$depbase.Tpo" -c -o applications/imap4client.
o applications/imap4client.cpp;  then mv -f "$depbase.Tpo" "$depbase.Po"; else r
m -f "$depbase.Tpo"; exit 1; fi
depbase=`echo applications/imap4server.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`;  i
f g++ -DHAVE_CONFIG_H -I. -I. -I../include  -ICrypto-PAn.1.0  -g -O2 -MT appli
cations/imap4server.o -MD -MP -MF "$depbase.Tpo" -c -o applications/imap4server.
o applications/imap4server.cpp;  then mv -f "$depbase.Tpo" "$depbase.Po"; else r
m -f "$depbase.Tpo"; exit 1; fi
depbase=`echo applications/tlsparser.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`;  if
g++ -DHAVE_CONFIG_H -I. -I. -I../include  -ICrypto-PAn.1.0  -g -O2 -MT applica
tions/tlsparser.o -MD -MP -MF "$depbase.Tpo" -c -o applications/tlsparser.o appl
ications/tlsparser.cpp;  then mv -f "$depbase.Tpo" "$depbase.Po"; else rm -f "$d
epbase.Tpo"; exit 1; fi
depbase=`echo applications/tlssession.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`;  if
 g++ -DHAVE_CONFIG_H -I. -I. -I../include  -ICrypto-PAn.1.0  -g -O2 -MT applic
ations/tlssession.o -MD -MP -MF "$depbase.Tpo" -c -o applications/tlssession.o a
pplications/tlssession.cpp;  then mv -f "$depbase.Tpo" "$depbase.Po"; else rm -f
 "$depbase.Tpo"; exit 1; fi
g++  -g -O2   -o netsniff -LCrypto-PAn.1.0 application.o parameters.o  pcapdev.
o uchar_traits.o  anonymization/ipanonymization.o  anonymization/stringanonymize
r.o  protocols/nullpacket.o protocols/etherpacket.o  protocols/ppppacket.o  prot
ocols/pppoediscoverypacket.o  protocols/pppoesessionpacket.o  protocols/arppacke
t.o protocols/ippacket.o  protocols/icmppacket.o protocols/tcppacket.o  protocol
s/udppacket.o protocols/dnspacket.o  tcpstreams/tcpdataflow.o  tcpstreams/tcpstr
eam.o  tcpstreams/tcpstreammanager.o  applications/appparser.o  applications/deb
ugparser.o  applications/httpparser.o  applications/ftpparser.o  applications/sm
tpparser.o  applications/pop3parser.o  applications/imap4parser.o  applications/
imap4common.o  applications/imap4client.o  applications/imap4server.o  applicati
ons/tlsparser.o  applications/tlssession.o -lCryptoPAn -lpcap -lcrypto
Leaving src
Making all in include
make  all-am
Leaving include
Making all in scripts
Leaving scripts
jbut:~/Development/ice/netsniff>
```

Autoconf

- Input files
  - *configure.(aclin)*
    - List of macros to scan and check on system
    - List of Makefiles to generate
  - *aclocal.m4, acsite.m4*
    - Set of M4 macros that can be used in the *configure.(aclin)* file that are not part of the standard **autoconf** macro set
- Output files
  - *configure*
    - Script to execute to build the *Makefile(s)*

SWINBURNE UNIVERSITY OF TECHNOLOGY

Autoheader

- Input files – same as **autoconf**
- Output files
  - *config.h.in*
    - Used as input when running configure to generate config.h

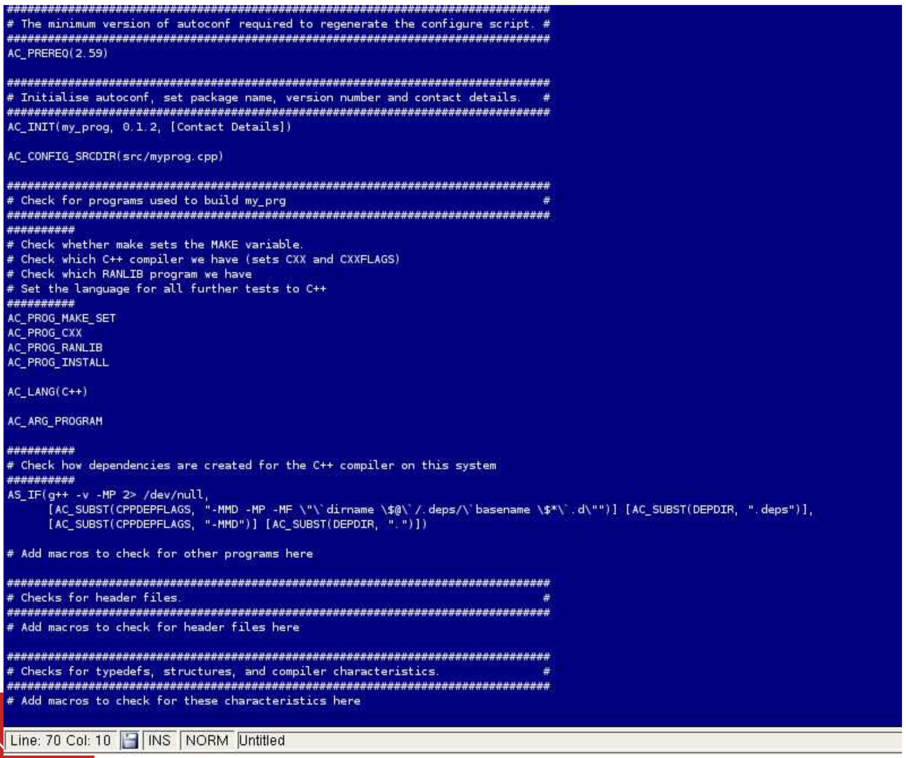
SWINBURNE UNIVERSITY OF TECHNOLOGY

Autoconf

- Advantages
  - Pre-existing macro set to check for existence of:
    - Tools
    - Programs
    - Libraries
    - Headers
  - M4 Macro language
  - Can put shell script into configure.(aclin)
  - Can be used without **automake**
- Disadvantages
  - M4 Macro Language – need to learn yet another language
  - Remembering cycle of applications to run to properly regenerate all required files

SWIN BUR NE SWINBURNE UNIVERSITY OF TECHNOLOGY

Autoconf – *configure.in* sample



```
################################################################################
# The minimum version of autoconf required to regenerate the configure script. #
################################################################################
AC_PREREQ(2.59)

################################################################################
# Initialise autoconf, set package name, version number and contact details.   #
################################################################################
AC_INIT(my_prog, 0.1.2, [Contact Details])

AC_CONFIG_SRCDIR(src/myprog.cpp)

################################################################################
# Check for programs used to build my_prg                                      #
################################################################################
##########
# Check whether make sets the MAKE variable.
# Check which C++ compiler we have (sets CXX and CXXFLAGS)
# Check which RANLIB program we have
# Set the language for all further tests to C++
##########
AC_PROG_MAKE_SET
AC_PROG_CXX
AC_PROG_RANLIB
AC_PROG_INSTALL

AC_LANG(C++)

AC_ARG_PROGRAM

##########
# Check how dependencies are created for the C++ compiler on this system
##########
AS_IF(g++ -v -MP 2> /dev/null,
      [AC_SUBST(CPPDEPFLAGS, "-MMD -MP -MF \"\`dirname \$@\`/.deps/\`basename \$*\`.d\"")] [AC_SUBST(DEPDIR, ".deps")],
      [AC_SUBST(CPPDEPFLAGS, "-MMD")] [AC_SUBST(DEPDIR, ".")])

# Add macros to check for other programs here

################################################################################
# Checks for header files.                                                     #
################################################################################
# Add macros to check for header files here

################################################################################
# Checks for typedefs, structures, and compiler characteristics.               #
################################################################################
# Add macros to check for these characteristics here
```

SWIN BUR NE SWINBURNE UNIVERSITY OF TECHNOLOGY

Using Autoconf without Automake

- Need to write our own set of *Makefile.in(s)*
- More effort
- Greater care needed in writing to ensure compatibility
- Resultant *Makefile(s)* are as neat or messy as the source *Makefile.in* templates

Not AutoMake (NAM)

- What is NAM
  - Basically a set of files that implements core build functionality in a way that minimises the effort involved in writing a *Makefile.in* file
  - Allows use of **autoconf** without **automake** AND simple generation of *Makefile.in*
  - Based in spirit on the WINE setup which uses **autoconf** but not **automake**
- Why NAM
  - Nicer *Makefile(s)* and make output
  - Re-usable

NAM – What does it Offer

- Default targets – all, clean, install, uninstall
- Recursive make in subdirectories
- C++ compilation
- Optional clean or verbose output during build
- Automatic dependency regeneration
- Automatic rerunning of **autoconf** and *configure* if necessary
- Readable Makefiles
- Linking or C++ archives and executables
- Installation of executable in $(prefix)/bin and $(prefix)/sbin
- Installation of **man** pages

SWINBURNE UNIVERSITY OF TECHNOLOGY

NAM – Required Files

- *NAM_rules.mk.in*
  - Common build rules
  - Configure generates NAM_rules.mk
    - Contains platform dependencies
    - Included into your Makefiles
- *bsd.mk*
  - BSD make specific instructions
- *gnu.mk*
  - GNU make specific instructions

SWINBURNE UNIVERSITY OF TECHNOLOGY

NAM – Template Files

- *configure.in*
  - ☐ Minimal set of autoconf macros required to generate a NAM compatible project
  - ☐ Need to add extra tests and variables as per your project requirements
- *Makefile.in*
  - ☐ Sample Makefile.in with all possible options for NAM
  - ☐ Remove unrequired functionality
  - ☐ Add extra and new compile rules

Creating NAM projects

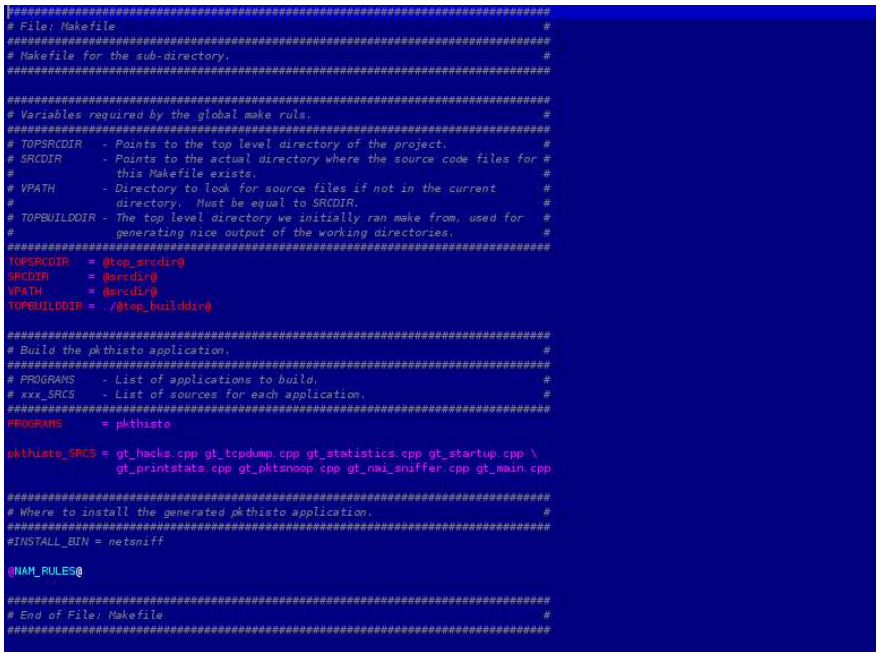
- Easier for a new project
  - ☐ Use the template *configure.in* and *Makefile.in* files and add to them as the project evolves
- More complex for an existing project
  - ☐ Use the template files and try to port macros from existing configure.in file – not **automake** macros
  - ☐ Add rules to *Makefile.in* as needed
  - ☐ Possibly extend *NAM_rules.mk.in* with new default rule (and submit changes back to me)

Example – pkthisto

- Recently converted **pkthisto** to use NAM
  - Existing package did not compile on FreeBSD 5.3
- Used default configure.in template
  - Added tests for libraries and header files used by **pkthisto**
- Developed *Makefile.in* to compile **pkthisto**
  - Link (and install) a single executable
  - List source files involved

SWIN BUR NE SWINBURNE UNIVERSITY OF TECHNOLOGY

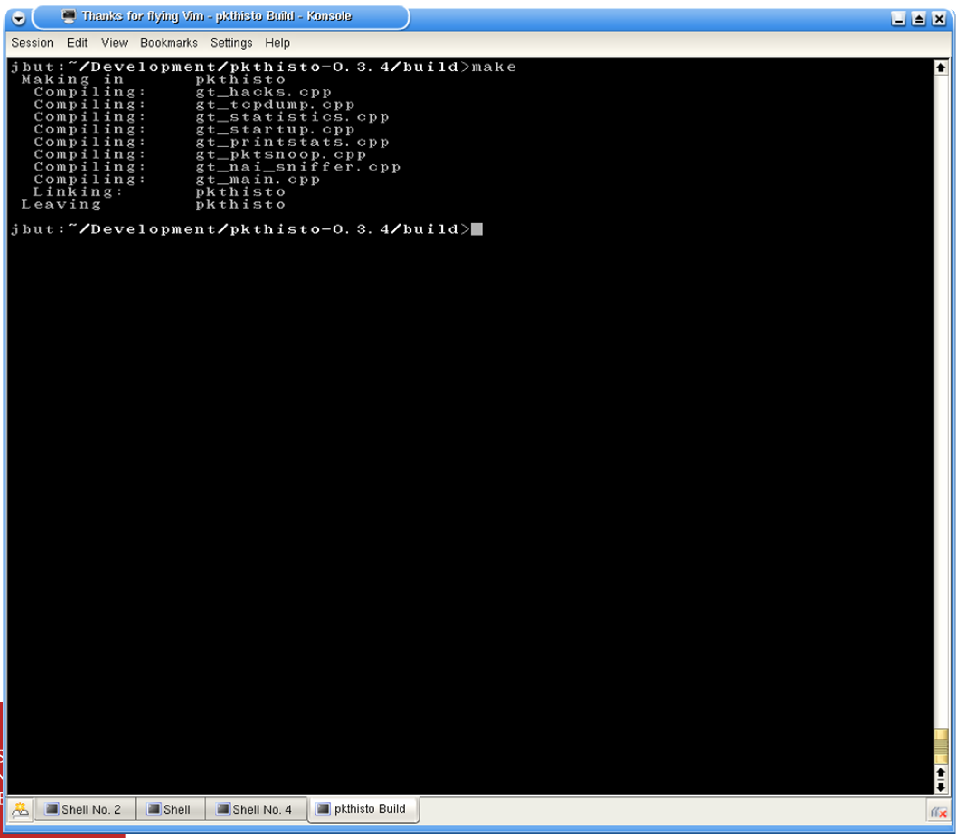
Example – pkthisto (Makefile.in)



```
###############################################################################
# File: Makefile                                                              #
###############################################################################
# Makefile for the sub-directory.                                             #
###############################################################################

###############################################################################
# Variables required by the global make ruls.                                 #
###############################################################################
# TOPSRCDIR   - Points to the top level directory of the project.             #
# SRCDIR      - Points to the actual directory where the source code files for #
#               this Makefile exists.                                          #
# VPATH       - Directory to look for source files if not in the current      #
#               directory.  Must be equal to SRCDIR.                           #
# TOPBUILDDIR - The top level directory we initially ran make from, used for  #
#               generating nice output of the working directories.            #
###############################################################################
TOPSRCDIR   = @top_srcdir@
SRCDIR      = @srcdir@
VPATH       = @srcdir@
TOPBUILDDIR = ./@top_builddir@

###############################################################################
# Build the pkthisto application.                                             #
###############################################################################
# PROGRAMS    - List of applications to build.                                #
# xxx_SRCS    - List of sources for each application.                         #
###############################################################################
PROGRAMS      = pkthisto

pkthisto_SRCS = gt_hacks.cpp gt_tcpdump.cpp gt_statistics.cpp gt_startup.cpp \
                gt_printstats.cpp gt_pktsnoop.cpp gt_nai_sniffer.cpp gt_main.cpp

###############################################################################
# Where to install the generated pkthisto application.                        #
###############################################################################
#INSTALL_BIN = netsniff

@NAM_RULES@

###############################################################################
# End of File: Makefile                                                       #
###############################################################################
```

SWIN BUR NE SWINBURNE UNIVERSITY OF TECHNOLOGY

Example – pktHisto (output)


Thanks for flying Vim - pkthisto Build - Konsole
Session Edit View Bookmarks Settings Help
Shell No. 2
Shell
Shell No. 4
pkthisto Build

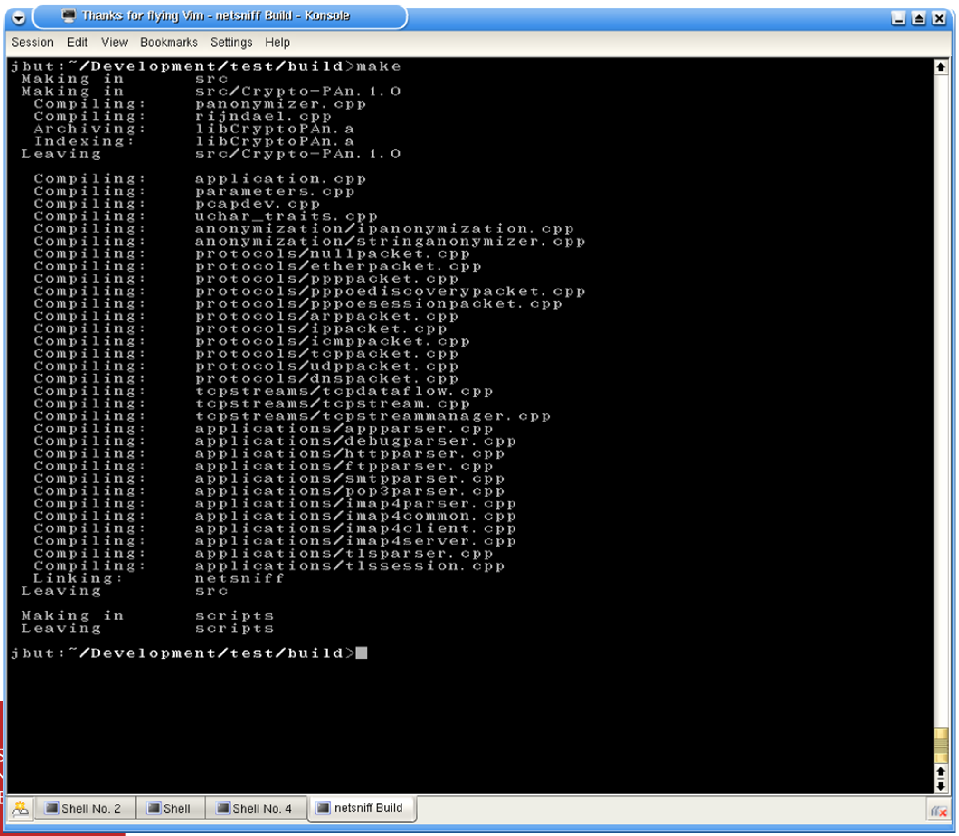
```
jbut:~/Development/pkthisto-0.3.4/build>make
 Making in        pkthisto
  Compiling:      gt_hacks.cpp
  Compiling:      gt_tcpdump.cpp
  Compiling:      gt_statistics.cpp
  Compiling:      gt_startup.cpp
  Compiling:      gt_printstats.cpp
  Compiling:      gt_pktsnoop.cpp
  Compiling:      gt_nai_sniffer.cpp
  Compiling:      gt_main.cpp
  Linking:        pkthisto
 Leaving          pkthisto

jbut:~/Development/pkthisto-0.3.4/build>
```

Example – netsniff

- Even though a recent project, netsniff compilation has evolved
  - Initially a single *Makefile* that built the source with nice output
  - Converted (by Urs) to use **autoconf** and **automake**
  - Converted (by me) to use **autoconf** and NAM
- More complex
  - Number of subdirectories
  - Temporary archive libraries
  - More **autoconf** tests to run
  - More configure options enabled

SWINBURNE UNIVERSITY OF TECHNOLOGY

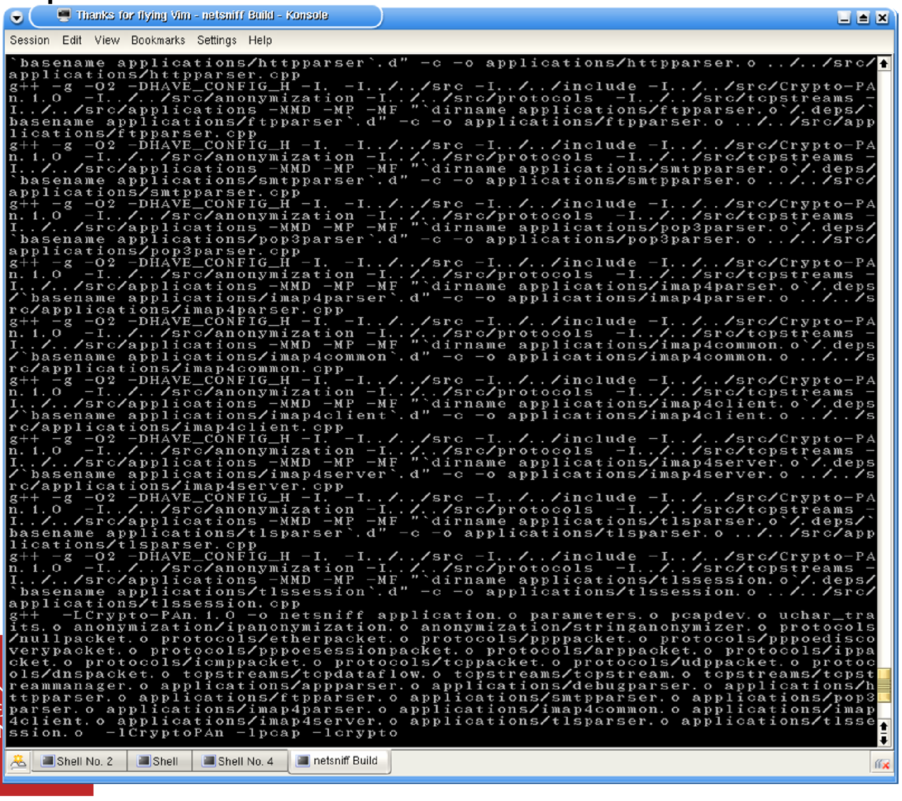
Example – netsniff (clean output)



```
jbut:~/Development/test/build>make
 Making in            src
 Making in            src/Crypto-PAn.1.0
  Compiling:          panonymizer.cpp
  Compiling:          rijndael.cpp
  Archiving:          libCryptoPAn.a
  Indexing:           libCryptoPAn.a
 Leaving              src/Crypto-PAn.1.0

  Compiling:          application.cpp
  Compiling:          parameters.cpp
  Compiling:          pcapdev.cpp
  Compiling:          uchar_traits.cpp
  Compiling:          anonymization/ipanonymization.cpp
  Compiling:          anonymization/stringanonymizer.cpp
  Compiling:          protocols/nullpacket.cpp
  Compiling:          protocols/etherpacket.cpp
  Compiling:          protocols/ppppacket.cpp
  Compiling:          protocols/pppoediscoverypacket.cpp
  Compiling:          protocols/pppoesessionpacket.cpp
  Compiling:          protocols/arppacket.cpp
  Compiling:          protocols/ippacket.cpp
  Compiling:          protocols/icmppacket.cpp
  Compiling:          protocols/tcppacket.cpp
  Compiling:          protocols/udppacket.cpp
  Compiling:          protocols/dnspacket.cpp
  Compiling:          tcpstreams/tcpdataflow.cpp
  Compiling:          tcpstreams/tcpstream.cpp
  Compiling:          tcpstreams/tcpstreammanager.cpp
  Compiling:          applications/appparser.cpp
  Compiling:          applications/debugparser.cpp
  Compiling:          applications/httpparser.cpp
  Compiling:          applications/ftpparser.cpp
  Compiling:          applications/smtpparser.cpp
  Compiling:          applications/pop3parser.cpp
  Compiling:          applications/imap4parser.cpp
  Compiling:          applications/imap4common.cpp
  Compiling:          applications/imap4client.cpp
  Compiling:          applications/imap4server.cpp
  Compiling:          applications/tlsparser.cpp
  Compiling:          applications/tlssession.cpp
  Linking:            netsniff
 Leaving              src

 Making in            scripts
 Leaving              scripts

jbut:~/Development/test/build>
```

Example – netsniff (verbose output)

- Output of "`make VERBOSE=2`"



```
`basename applications/httpparser`.d" -c -o applications/httpparser.o ../../src/
applications/httpparser.cpp
g++ -g -O2 -DHAVE_CONFIG_H -I. -I../../src -I../../include -I../../src/Crypto-PA
n.1.0  -I../../src/anonymization -I../../src/protocols  -I../../src/tcpstreams -
I../../src/applications -MMD -MP -MF "`dirname applications/ftpparser.o`/.deps/`
basename applications/ftpparser`.d" -c -o applications/ftpparser.o ../../src/app
lications/ftpparser.cpp
g++ -g -O2 -DHAVE_CONFIG_H -I. -I../../src -I../../include -I../../src/Crypto-PA
n.1.0  -I../../src/anonymization -I../../src/protocols  -I../../src/tcpstreams -
I../../src/applications -MMD -MP -MF "`dirname applications/smtpparser.o`/.deps/
`basename applications/smtpparser`.d" -c -o applications/smtpparser.o ../../src/
applications/smtpparser.cpp
g++ -g -O2 -DHAVE_CONFIG_H -I. -I../../src -I../../include -I../../src/Crypto-PA
n.1.0  -I../../src/anonymization -I../../src/protocols  -I../../src/tcpstreams -
I../../src/applications -MMD -MP -MF "`dirname applications/pop3parser.o`/.deps/
`basename applications/pop3parser`.d" -c -o applications/pop3parser.o ../../src/
applications/pop3parser.cpp
g++ -g -O2 -DHAVE_CONFIG_H -I. -I../../src -I../../include -I../../src/Crypto-PA
n.1.0  -I../../src/anonymization -I../../src/protocols  -I../../src/tcpstreams -
I../../src/applications -MMD -MP -MF "`dirname applications/imap4parser.o`/.deps
/`basename applications/imap4parser`.d" -c -o applications/imap4parser.o ../../s
rc/applications/imap4parser.cpp
g++ -g -O2 -DHAVE_CONFIG_H -I. -I../../src -I../../include -I../../src/Crypto-PA
n.1.0  -I../../src/anonymization -I../../src/protocols  -I../../src/tcpstreams -
I../../src/applications -MMD -MP -MF "`dirname applications/imap4common.o`/.deps
/`basename applications/imap4common`.d" -c -o applications/imap4common.o ../../s
rc/applications/imap4common.cpp
g++ -g -O2 -DHAVE_CONFIG_H -I. -I../../src -I../../include -I../../src/Crypto-PA
n.1.0  -I../../src/anonymization -I../../src/protocols  -I../../src/tcpstreams -
I../../src/applications -MMD -MP -MF "`dirname applications/imap4client.o`/.deps
/`basename applications/imap4client`.d" -c -o applications/imap4client.o ../../s
rc/applications/imap4client.cpp
g++ -g -O2 -DHAVE_CONFIG_H -I. -I../../src -I../../include -I../../src/Crypto-PA
n.1.0  -I../../src/anonymization -I../../src/protocols  -I../../src/tcpstreams -
I../../src/applications -MMD -MP -MF "`dirname applications/imap4server.o`/.deps
/`basename applications/imap4server`.d" -c -o applications/imap4server.o ../../s
rc/applications/imap4server.cpp
g++ -g -O2 -DHAVE_CONFIG_H -I. -I../../src -I../../include -I../../src/Crypto-PA
n.1.0  -I../../src/anonymization -I../../src/protocols  -I../../src/tcpstreams -
I../../src/applications -MMD -MP -MF "`dirname applications/tlsparser.o`/.deps/`
basename applications/tlsparser`.d" -c -o applications/tlsparser.o ../../src/app
lications/tlsparser.cpp
g++ -g -O2 -DHAVE_CONFIG_H -I. -I../../src -I../../include -I../../src/Crypto-PA
n.1.0  -I../../src/anonymization -I../../src/protocols  -I../../src/tcpstreams -
I../../src/applications -MMD -MP -MF "`dirname applications/tlssession.o`/.deps/
`basename applications/tlssession`.d" -c -o applications/tlssession.o ../../src/
applications/tlssession.cpp
g++  -LCrypto-PAn.1.0 -o netsniff application.o parameters.o pcapdev.o uchar_tra
its.o anonymization/ipanonymization.o anonymization/stringanonymizer.o protocols
/nullpacket.o protocols/etherpacket.o protocols/ppppacket.o protocols/pppoedisco
verypacket.o protocols/pppoesessionpacket.o protocols/arppacket.o protocols/ippa
cket.o protocols/icmppacket.o protocols/tcppacket.o protocols/udppacket.o protoc
ols/dnspacket.o tcpstreams/tcpdataflow.o tcpstreams/tcpstream.o tcpstreams/tcpst
reammanager.o applications/appparser.o applications/debugparser.o applications/h
ttpparser.o applications/ftpparser.o applications/smtpparser.o applications/pop3
parser.o applications/imap4parser.o applications/imap4common.o applications/imap
4client.o applications/imap4server.o applications/tlsparser.o applications/tlsse
ssion.o  -lCryptoPAn -lpcap -lcrypto
```

Using NAM

- Installing the required files
  - Obtain *NAM_rules.mk.in*, *bsd.mk* and *gnu.mk* and place a copy in the top directory of your project
  - *NAM_rules.mk* will be generated in the top build directory after running **configure**

Using NAM

- Writing *Makefile.in(s)*
  - Obtain the template file *Makefile.in* and place a copy in the top directory of your project AND in each subdirectory you wish to recursively make
  - The first four lines of the Makefile.in file are mandatory and MUST NOT be commented out or deleted

```
TOPSRCDIR   = @top_srcdir@
SRCDIR      = @srcdir@
VPATH       = @srcdir@
TOPBUILDDIR = ./@top_builddir@
```

Using NAM

- Writing *Makefile.in(s)* – variables
  - ☐ **SUBDIRS** – list of subdirectories for make to recurse into
  - ☐ **INCLUDES** – compiler flags to list extra directories to search for included files
  - ☐ **PROGRAMS** – list of executables to link
  - ☐ **ARCHIVES** – list of temporary library archives (.a) to link
  - ☐ **xxx_SRCS** – list of C++ source files to compile in order to build program **xxx**, where **xxx** is a program in the **PROGRAMS** or **ARCHIVES** variable
  - ☐ **xxx_LIBS** – list of libraries to use when linking the program or archive xxx
  - ☐ **xxx_LDFLAGS** – linker flags to use when linking the program or archive xxx
  - ☐ **PRECOMP_HEADER** – list of header files to compile using precompiled headers (requires gcc3.4+)
  - ☐ **INSTALL_BIN** – list of executables to install to $(prefix)/bin
  - ☐ **INSTALL_SBIN** – list of executables to install to $(prefix)/sbin
  - ☐ **INSTALL_MAN** – list of man pages to install to $(prefix)/man

SWINBURNE UNIVERSITY OF TECHNOLOGY

Using NAM

- Writing *Makefile.in(s)*
  - ☐ Do NOT delete the line

    ```
    @NAM_RULES@
    ```
  - ☐ This includes the rules defined in NAM_rules.mk
  - ☐ Add any new and other rules AFTER the **@NAM_RULES@** line
    - ☐ Can add extra dependencies for all, clean, install and uninstall targets
    - ☐ New targets (all, install, uninstall) with rules will be executed AFTER the default make of these targets
    - ☐ Any rules before **@NAM_RULES@** will supercede make all as the default target

SWINBURNE UNIVERSITY OF TECHNOLOGY

Using NAM

- Writing *configure.in*
  - Do not remove any existing macros from this file
  - Add new macros to test for anything you need where specified in the file
  - Add a list of all Makefiles your project needs to the **AC_CONFIG_FILES** macro, you MUST ensure that your *Makefile(s)* are listed AFTER the *NAM_rules.mk* file already there
    - Otherwise your *Makefiles* will be generated with the old *NAM_rules.mk* and will be one configure cycle out of step

Using NAM

- Writing *configure.in*
  - Source for writing *configure.in* tests
  - Download the **autoconf** manual from http://www.gnu.org/software/autoconf/manual/index.html
  - **Autoconf** website – http://www.gnu.org/software/autoconf
  - Google search for help
  - Using the minimal *configure.in* will not be a problem, it just means that occasionally make will fail where it would be better if *configure* failed – the idea is if configure completed successfully, the system is capable of building the application

Conclusions

- Concept behind the **autotools** suite is good
  - **Autoconf** is well implemented, **automake** is a mess
  - NAM allows use of **autoconf** and minimal work in writing *Makefile.in* files
- NAM takes advantage of a prewritten rule set to minimise work on the build environment
  - Leverage my 8 weeks effort into learning how autoconf works
  - Get readable Makefiles and build output with no effort
- Easy to use, especially for new projects
- Make your software development platform independent from the start
  - Lets you run tests at home if you have a different platform (such as Linux or MacOS
  - Lets your software be used by others

SWINBURNE UNIVERSITY OF TECHNOLOGY

Questions

- And the title says it all…

SWINBURNE UNIVERSITY OF TECHNOLOGY