Reverse Engineering etc.
Sep 14, 2009

Topics

- Adminisitrivia
- Basic Tool Overview
- Reverse Engineering Overview
- GDB in depth
- Windows tools

Administrivia

The dreaded H1N1

- If you think you're sick, don't come to class.
  - Email myself and the TA's
  - If you're going to be out for more than two classes, please get a doctor's note.
  - If you're going to miss an exam, get a doctor's note.
  - Labs will continue to be available online as well as notes.
  - TA's and I are available via email and phone.
- The key here is communications. Keep us appraised as to your condition and status.

Administrivia

Lab 1 Erratum and Advice

- Change LONG_MIN & LONG_MAX to INT_MIN & INT_MAX respectively. This will fix the problem where the code thinks tmin should return 0. (It will also cause the default bits.c to generate lots of errors in btest.)
- You should change (team_check=1 to team_check=0 in btest.c). You can also run "btest -a" to supress the check at runtime (that's what our test script does).
- You might want to run "btest -f [function name]" to focus on testing one function at a time. For example, "btest -f fitsBits" just tests the fitsBits function. This prevents you from having to sort through all the errors for functions you haven't implemented yet.
- Write out what the answers should be and look for patterns.
- No conditional tricks needed for these problems.

Tool Overview

SSH – Secure login and copies

Emacs—Editor of all things

Make—Save time building large and not so large projects

CVS—Source repository that will save your gluteus maximus on more than one occasion.

GCC—Compiler of lingua de jour

GDB—Debugger extraordinaire

SSH

Two different ways to log in with SSH

- **Pre-shared-key (PSK) : Use your directory password**
  - **ssh <yourdirectoryid>@linux.grace.umd.edu**
- **Public key: Use only ONE password for all of your accounts (sort of)**
  - **On your local machine: ssh-keygen –t rsa –b 2048**
  - **On a grace machine:**
    - mkdir ~/.ssh
    - pwd (note what is printed out)
  - **On your local machine:**
    - scp ~/.ssh/id_rsa.pub <yourdirectoryid>@linux.grace.umd.edu:/<yourhomedir>/.ssh/authorized_keys
    - ls –l on grace to ensure ONLY you can write to authorized keys
  - **Tunneling X windows: use –X option to ssh.**

EMACS

Yes it isn't the easiest to learn (ctrl-meta-X)

- But-it pretty much will work everywhere. Key bindings for Eclipse etc.
- Really useful when you only have a dumb terminal connection, i.e. ssh without an X window tunnel

Basic Movement

- Forward one char: ctrl-f
- Backward one char: ctrl-b
- Forward one word: meta-f, meta-b (for backwards)
- First char of line: ctrl-a, ctrl-e (end of line)

File operations

- Open: ctrl-x-ctrl-f
- Save: ctrl-x-ctrl-s
- Revert: meta-x "revert buffer"

EMACS continued

- Modes
  - C mode
  - ASM mode
- GDB mode
  - Meta-x gdb

Makefiles (gmake only)

- **Example Makefile line**

```
all: foo
    $(CC) $(CFLAGS) foo.c –o foo

all: is target
foo is dependency
$(CC) …. is action
```

- **Rules**

```
%.tex: %.dvi
    Latex $<
```

- **Phony targets used when no file created**

```
clean:
    rm –rf $(OBJS) $(DEPS)
.PHONY: clean
```

Makefiles

- **Makefile tricks**

  SOURCES := $(wildcard *.c)

  OBJ := $(patsubst %.c, %.o, $(SOURCES)

- **Automatic Dependencies**
  - Add –MD flag to CFLAGS

  DEPS := $(patsubst %.o, %.d, $(OBJS))

  .include $(DEPS) /* at the end of your file */

CVS

- Starting a new repository
  - Cvs –d <PATH> init
- Set your environment variables
  - Export CVSROOT=<path>
  - Export CVS_RSH=/usr/local/bin/ssh
- Start a new project by importing source
  - cvs import –m “Initial sources” myprojectname <yourid> start
- Checkout source from an archive
  - cvs checkout myprojectname
- Updating (only useful for multiple people editing)
  - cvs update

CVS continued

- Comparing versions
  - cvs diff (will show you the delta between current and repos)
- Committing changes
  - cvs commit
- Reverting if committed a bad version
  - cvs update –j <current rev#> -j <revert rev#> filename
- Tags
  - cvs –q tag Working-at-3am
  - cvs commit
  - Cvs checkout –r Working-at-3am <projname?

CVS continued

- Keyword substitution fun
  - Put in comments at head of file
    - $Revision$ - replaces with current revision number
    - $Header$ - filename and revision number
    - $Author$ - replaces with the author id (checking in)
    - $Date$
    - Log - a pushdown of CVS Log entries
- More help?
  - http://cvsbook.red-bean.com/cvsbook.html

GCC

- **Good options to know**
  - **-o <output file name>** ***rather than default a.out***
  - **-c** ***compile but don't link***
  - **-g** ***produce debugging info***
  - **-O** ***optimize code***
  - **-S** ***produce assembler***
  - **-E** ***only preprocess***
  - **-MD** ***produce dependency info files***

Reverse Engineering

- Legal Issues
  - A number of laws limit what you can do with RE
    - UCITA
    - DMCA
  - Always consult management/lawyers before doing RE at work
- White box testing
  - Source code available
  - Trying to understand source and function of program
- Black box testing
  - No source available
  - Want to understand function of program
  - Perhaps for interoperability
  - Disassemble/decompile
  - Test inputs and outputs

GDB

- A good intro tutorial
  - http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html
- Cheat sheet for GDB
  - http://www.digilife.be/quickreferences/QRC/GDB%20Quick%20Reference.pdf
- Demo

Reversing Strategies

- Strings
- Set break points and follow flow
- Disassemble important blocks and functions
- Demo