An overview of some tools for programming in the UNIX environment.

Abstract

September 12, 2000

Robert Spier

Overview of Programming in the UNIX Environment
This Presentation

Overview of Programming in the Unix Environment

September 12, 2000

Then, use the newsgroups.

For more information, first try the resources in the next section.

We are going to go through some things very quickly.

This is a whirlwind tour.

Overview of lots of things.
toolset.

● UNIX programmers who were forced to work on Windows ported the established

● Most of these tools/techniques can also be used there.

Much of this applies to the Windows Environment as well.

Windows
but also some really big programs too

contains lots of small programs used together

is infinitely flexible

The Unix Environment
This talk...

Other Tools

Debugging

Compiling

Manipulating Files

Getting Help
There are many resources already on the systems:

- Answers
- Reference cards (grep, emacs, etc.)

- CETS
- Other documentation
- Zinfo documents

- http://tiling.seas.cwru.edu/man.8t
- man pages

Getting Help
man pages are organized into sections and subsections

Finding man pages

1. Commands

2. System services and error numbers

3. User-level library functions

4. Device drivers, protocols, and network interfaces

5. File formats used or read by various programs

6. Games and demos

7. Miscellaneous useful information pages

8. System maintenance and operation commands

Overview of Programming in the Unix Environment

September 12, 2000
For example: man -s 3s print

Use man -s section page to pick a specific one.

Print (3s) - print formatted output

Print (3p) - formatted output conversion

Print (1) - write formatted output

Specify the section. For example, print has 3 different man pages.

A named page can exist in more than one section, so sometimes you need to

Finding man pages (2)
Finding man pages (3)

use man-1 keyword to perform a keyword search
Also, make sure you are reading the right pages.

Give up after the first reading.

Sometimes the man pages and documentation can be quite confusing.

Don’t

man page confusion
- use defja.com to search archives

   NewsGroups

- Google hitp://www.google.com

   The Web

answered your question already.

There is a ton of information out there. Other people may have asked or

Other sources of information
Manipulating Files

- Editing Files
- Moving Files Around
chmod, chgrp for changing permissions

more/less display a file page by page

cat display a file on your terminal

rm delete files

mv move/rename files

cp copy files

Moving Files Around
Efficient editor provides. Being able to fully utilize your editor will make you much more efficient. It is important to take the time to get to know what tools your editor provides. Generally, source code or editing files will spend most of your time editing files.
— does not support indentation, colors, etc.

- it is not a source code editor
- it is only designed for email
- don't even think about using pico
http://www.vim.org

pack/vim-5.5/bin/gvim gui

pack/vim-5.5/bin/vim command line version

but - I don't use it, so I can't tell you too much more.

- supports syntax highlighting, indentation, and other advanced things.

- use vim, VIM improved, instead

- small, fast, and arcane

- vi is related to ex, which is related to ed

vi
Towards the end of the lecture, we are going to go into great detail about

how to use Emacs.

Towards the end of the lecture, we are going to go into great detail about

that with friends' or browse the web.

environment, that will let you do everything from edit code, to read your mail;

it is an exception to the unix paradigm of small programs. Emacs is a complete

my recommendation for a editor

Emacs

September 12, 2000
gcc -o hello hello.c

You should have learned the general concepts here in CSE240.

- Planning, and #2 - coding is compiling.
- The third most time consuming part of any programming project (behind #1

Compiling

Overview of Programming in the Unix Environment
September 12, 2000
Solution: use make

- Just recompile the one that changed, and re-link
- If only one changes, you don't need to recompile them all
- Often it is necessary to split programs across several source files

Make
A Simple Makefile: (incorrect indentation)

to be named Makefile,

make is a program which operates on files called Makefiles. These files need
document on the 386 web page, and the make documentation.
theMakefiles. For more info, please see the "Getting Started with The Makefile" Make and Makefiles are a topic large enough for an entire presentation by

Make (2)
Parts of the Makefile

- Shell Lines
- Dependency Lines
- Macro Lines
Another Makefile

Overview of Programming in the Unix Environment

September 12, 2000

rm *
clean:

data.o : data.c data.h
main.o : main.c data.h

# Dependencies for object files

> $ $(CC) $(CFLAGS) -c $@
  % : %.
c

program : $(OBJS)
  $(CC) $(LDFLAGS) $(OBJS) -o program

CFLAGS = -Iw
LDFLAGS = -02
CC = /bin/cc

OBJS = main.o data.o 10.o # All object files

# Makefile for the OS Project

#
The C pre-processor (cpp) is responsible for including the include files and dealing with other pre-processor directives. All pre-processor directives must be at the beginning of a line, and they start with # (pound, hash, sharpe, or octothorpe).

1. Preprocessing
2. Compilation
3. Assembly
4. Linking

The compilation process has several stages:
Substitution with `#define`
defines are often used for numerical constants (i.e., pi or error codes).

That would have been a pain to type. This also makes it easier to change things.

```c
{ ( 10 > -22 ) : ( 10 < -22 ) },
printf("ABS of 10 is %d\nABS of -22 is %d\n",
    printf("pi is %f\n", M_PI),
    printf("Hello World!\n"));
}

void main()
```

becomes...

**Substitution with #define (2)**
Only if debugging

Always!

$s .a .out
$.

&gcc -DDEBUGGING appmain.c

Gives us...

{
#endit
printf("Only if debugging\n");

define IMEBUGGING

printf("Always!\n");
}

void main()

Conditional Compilation
- choose output file
- stop after compilation, do not assemble
- only run preprocessor
- optimize
- include debugging info
- enable all warnings
- enable common warnings

fcc flags
gcc flags (2)

- l add to library search path
- l link in a specific library
- I add to include <> file path
- c compile
. 0 Object file; passed to linker

.1 Preprocessor file; not usually named on command line

.5 Assembler source; assemble

.c, .cc, .cxx C source; preprocess, compile, assemble

It is important to use consistent and proper file extensions with gcc.

File extensions
Practice it!

You will spend lots of time debugging. It makes sense to get good at it.

Debugging

September 12, 2000
bt - show backtrace
x - examine memory
p /fmt - print
hi/si - next instruction
s - next line, step in
u - next line, step over
help <thinity>
delete <breakpoint number>
b <line number>
b <function name>
b - set breakpoint

Important keystrokes:

`gdb` is the GNU Debugger, it works with gcc, the GNU C compiler.
`gdb`
It's hard to show how to use gdb with pictures, so we'll do a very short demonstration here.

8db demonstration
commands it is using when you click something, you can learn Gdb from it. The only downside to it is that it is slow. But since it shows you the Gdb powerful than the Visual Studio debugger. It is full featured, and more structures and how they relate to each other. It will allow you to visually see your data

**ddd: The Data Display Debugger**

September 12, 2000

Overview of Programming in the Unix Environment
(2) ddd
{ 
    puts("This\tcrash!\n")
    p = \n    int * malloc(100);
} void main
#include <stdio.h>
#include <malloc.h>

An obviously bad program, right?

It was originally developed at Pixar.

Electrical Fence helps you detect two common programming bugs: software that overruns the boundaries of a malloc() memory allocation, and software that touches a memory allocation that has been released by free().
Fix it now!

us later.

That's not right. We know that program was bad. This will come back to haunt.

Don't crash!

$ ./a.out
$ gcc engine.c

electric fence (2)
Electric Fence will be a lifesaver. Use it.

If you run that in gdb, it will stop on the line causing the problem.

Segmentation Fault

Electric Fence 2.0.5 Copyright (C) 1987-1998 Bruce Perens.

```
./a.out
$ gcc efence.c -I/home4/c/cse381/lib -lefence
$`
Time permitting: show a customized profile file here.

Change your shell. I recommend bash.

Use the bash program to

write either bash or tcsh. (Both have tab completion.) Unless you have a specific reason not to, you should make sure that you are

Eniac and the CIS machines have several different shells installed on them.

Your main interface to UNIX is the shell.

This is another topic which could fill an entire presentation...

Controlling / Configuring Your Environment

Overview of Programming in the Unix Environment

September 12, 2000
the logistics of this for you, but it is called job control.

environment. You can run more than one program at once. Your shell handles

You should know by now that UNIX is a multiprogramming / multitasking

Job Control

Overview of Programming in the Unix Environment
Tail is a super-powerful super simple tool. It searches through files line by line, returning things that match. Here, we pipe the output of grep into head and tail.

Toorc: piping, grep, head, tail
358 total
241 barbers
91 mutexes
26 common
$ wc -l [ch] sort

wc counts words, lines, and characters. sort, sorts.

$ zoom
$ zoom
$ zoom

grep also supports regular expressions:

tools: grep (2), wc, sort
RCS files are named the same as the file you are putting under version control, but with ‘v appended to it.

Changes, so if something breaks, you can track down what code changed.

It is good for keeping track of what you've done, and having a record of

documentation, graphics, papers, and form letters.

Revisions, RCS is useful for text that is revised frequently, for example programs.
RCS automates the storing, retrieving, identifying, and merging of

The Revision Control System (RCS) manages multiple revisions of files.

Source Control: RCS
rcsdiff displays the difference between version of the file. You can tell it to show the differences
between any two arbitrary versions. The -u specifies the format of the diff.

```bash
$ vim test.c
.done
INITIAL REVISION: 1.1
```

NOTE: THIS IS NOT the log message!
enter description, terminated with single \.
 or end of file:
test.c → -- test.c

```bash
cat -l test.c
{}
void main() { printf("Hello\n");
cat test.c
```

done with it, you should use cat -u to check it in unlocked.
cat -l checks in the current version and "locks" it for your use. When you are

rcs example

Overview of Programming in the Unix Environment
done

>>

>> changed from print to printf

enter 'Q' to terminate, terminated with single 'Q' or end of title.

new revision: 1.2; previous revision: 1.1

test.c, y -- test.c

c - u test.c

{ main() 
    { puts() }
    { printf() }
        
        "Hello World"
        
        ++ test.c 2000/09/11 02:42:14
-- test.c 2000/09/11 02:41:19

diff -u test.1 test.1

reverting revision 1.1
RCS file: test.c

=====================================================================
$c diff -u test.c

rcs example (continued)
wide area networks allows triggers to enable/log/control various operations and works well over a
distributed file system. Among multiple authors, CVS helps to manage releases of directories consisting of version controlled files. Unlike RCS or SCCS, CVS does not just operate on files (usually source code), keep a log of who, when, and why changes occurred.

Version Control: CVS

Overview of Programming in the Unix Environment
September 12, 2000
or your team, please ask a TA.

If you have any questions about which one might be more appropriate for you

with the repository periodically:

Information are stored. Each user has their own local copy, which gets synchronized

CVS works remotely. There is a central repository where the files and version

at a time.

files at once. RCS acts as the gatekeeper, allowing only one person to edit a file

RCS generally works “in place” (i.e., everyone is working on the same physical

RCS vs. CVS

Overview of Programming in the Unix Environment

September 12, 2000
More Emacs

About Emacs.

Time permitting... we do the rest of the slides which go into more detail.
Emacs also understands Esc followed by the key (\texttt{Esc k}) to be the same as \texttt{M-k}.

If your keyboard doesn't have a Metakey, use the Alt key. If that doesn't work,

\begin{itemize}
  \item \texttt{C-x C-k} \texttt{Control X} followed by \texttt{k}
  \item \texttt{C-x C-c} \texttt{Control C} followed by \texttt{Control c}
  \item \texttt{C-k} \texttt{Control k}
  \item \texttt{M-k} \texttt{Meta (Alt or Esc) k}
\end{itemize}

Emacs uses the following notation for keybindings:

\begin{itemize}
  \item \texttt{Escape Meta Alt Control Shift}
\end{itemize}
Tab completion is supported.

\texttt{M-x replace-regexp}

replace-regexp function, type:
means execute-extended-command. For example, if you wanted to call the
Emacs supports too many commands to have them all bound to keys. M-x

\texttt{Execute Extended Command}
Emacs GUI

If you are running Emacs in X or under Windows, there is a fully functional GUI with pull-down menus and dialog boxes.

bindings.

If you’re scared of Emacs already, don’t be. It is possible to just use a fraction of it’s features, and be quite happy. You don’t have to memorize all the key bindings.
Getting Help in Emacs

C-h a  apropos - search for a function

C-h f  help on a function

C-h k  help on a specific key

C-h m  help on this mode

C-h ?  help
C-x C-c prompt to save all files and then quit

C-x w write file

C-x s save file

C-x f find file (and load it into a buffer)

Loading and Saving Files
Elmacs also keeps track of your previous entries. You can use the up and down arrows or M-p and M-n) to scroll up and down through them.

Possible matches:

| Tab Completion allows you to type the first few characters of a filename, hit | Tab Completion allows you to type the first few characters of a filename, hit |

---

September 12, 2000

Overview of Programming in the Unix Environment
Buffers

- C-x C-b List buffers
- C-x k Kill buffer
- C-x b Select buffer

- Some can be running programs (shell, make) or internal things
- Most buffers are directly associated with files
- Emacs can have multiple buffers open
Emacs supports multiple windows inside a single frame.

```
----- common.h -----

#include "mutex.h"

void mutex_init(mutex_t *m);
void mutex_lock(mutex_t *m);

struct mutex {
    char lock;
} mutex_t;

/* mutex.c */

----- bpar.c -----**

#include <unistd.h>
#include <malloc.h>
#include <stdlib.h>
#include <stdio.h>
#include <signal.h>
#include <context.h>
#include <poll.h>

BUFFERS FILES Tools Edit Search Help C Help
```

Windows
Windows (2)
Frames (2)

(Staring multiple copies is a bad thing because of potential conflicts if you edit the same file in multiple copies. There is no reason to start multiple copies of emacs up. Frames allow you to have multiple emacs windows on the screen, all sharing the same frame in new frame)

C-x 5 0 delete frame

C-x 5 f visit file in new frame

C-x 5 b open buffer in new frame

C-x 5 2 duplicate this buffer in a new frame
Emacs has amazingly powerful search and replace tools.

search-bw    C-
search-f      C-
query-replace-regexp
replace-regexp
replace-string
query-replace
M-\%
You can usually manually switch modes by using M-x mode-name-mode.

When you open a .c file, it automatically sets the buffer to c-mode. Emacs also knows about the file name extension associated with each mode. So text-mode ●

C++-mode ●

c-mode ●
elisp-mode ●

code and variables. Sample modes include:

Emacs categorizes things based on modes. Each mode is a collection of elisp

Modes

September 12, 2000

Overview of Programming in the Unix Environment
you to jump directly to a function.

It shows you a list of files in the current directory, and allows you to click on them and switch between them. It also scans them for function names, and allows

most of the programming related modes (c-mode, java-mode).

Speedbar is a nifty little utility (mode) that is part of emacs. It works with

Speedbar (2)
font-lock-mode provides syntax highlighting. Syntax highlighting makes code easier to read and helps catch typos.
To compile/recompile

Goto-Line

We have provided two extra keybindings for you:

used for font-lock-mode. One thing you may wish to do is change the colors

We have provided you a sample .emacs file which sets some sensible and useful

Emacs is infinitely configurable using Elisp.

Configuring Emacs
#include <basic_vi_key_commands.txt>
Any questions?

The End