CS350 Lab0

Sunil Agham & Deepal Shah
OSNet C.S. Dept.

Outline

- Virtual System
- Linux Editors
- GCC
- GNU Make
- Debugging
- Getting Help

- Warmup Lab Activity

Virtual System

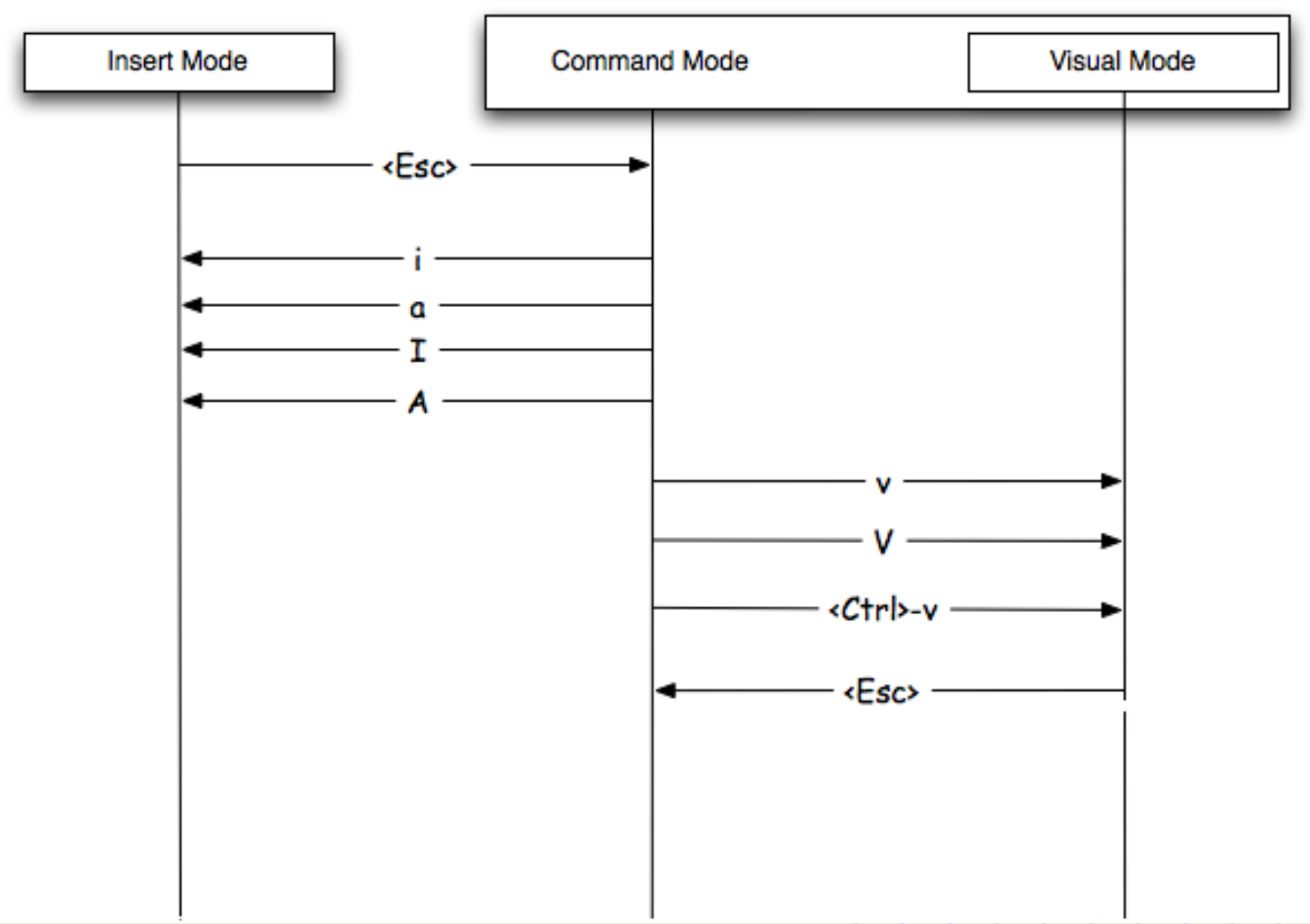
- Your own kernel, not local to physical machine
- Login instructions:
  - http://cs.binghamton.edu/~kartik/cs350/
  - on "boot:" prompt, enter name of your virtual system (your username)
- Initial root password: .virtual_access
  - Its a hidden file: use *"ls -a"*
  - Its at "/home/$USER". NOT at "/home/"
- On first usage:
  - change root password
  - create new user
  - use non-root user unless root access is needed

Linux Editors

- Vi & Vim (I prefer)
  - Vim (Vi IMproved)
  - Command line
  - Many powerful features (especially for programmers)
    - Multiple programming language support
    - Connection with cscope, ctags (code browsing tools)
    - Split screen
    - Search and replace
- Tutorial: http://blog.interlinked.org/tutorials/vim_tutorial.html

Linux Editors (cont..)

- Vim (cont..)



Linux Editors (cont..)

- Other Graphical Editors
  - kwrite
  - kate
  - gedit
  - kedit
- IDE
  - geany
    - argument list, standard library functions auto-suggest
    - compile and execute through tool (instead of CLI)
  - eclipse

GCC

- Originally: GNU C Compiler
- Later extended and now stands for GNU Compiler Collection
- Native-compile as well as Cross-compile
- Supports many different languages
- Many compiler options
  - "*-o file*" output file
  - "*-E*" stop after preprocessing
  - "*-c*" compile but don't link
- Examples:
  - gcc -o test test.c
  - gcc helloFunc.c helloMain.c -o hello

GNU Make

- For "Not so small" programs
  - Many programmers, Many Lines of code (Linux kernel 3.2: 14,998,651 lines)
  - Long files are harder to manage
    - longer compilation time
    - many programmers can't work simultaneously
- Solution
  - divide project into components
  - minimum compilation when something is changed
- Manual: http://www.gnu.org/software/make/manual/make.html

GNU Make (cont..)

- Format:
  target ... : dependencies ...
  command
  ....
  ....
- There should be a tab before command
- Example:
  hellomain:hellomain.c hellofunc.c
  gcc -o hello hellomain.c hellofunc.c
- To compile above example:
  $ make
  $ make hellomain

Debugging

- gdb
  - Add "-g" option in gcc (enable debugging support)
  - Run as "gdb <*executable_name*>"
  - gdb commands
    - run
    - break
    - continue
    - step
    - next
    - print
  - For setting input arguments: "set args"
  - With corefile: "gdb <*executable_name*> <*corefile*>"

Getting Help

- For most of unix / linux utilities
  - -h or --help
  - man pages
  - Google

Warmup Lab Activity (Lab 0)

- Recursive Fibonacci Sequence

Fibonacci Introduction

A sequence of numbers like this:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, ...

Formula of Fibonacci Sequence is:

$F(1) = 1$

$F(2) = 1$

$F(n) = F(n-1) + F(n-2)$

How to program this formula in C ?

Pseudo Code

Simple Recursive Algorithm:

```
int fib(int number) {
    if number is 1 or 2
        return 1;
    else
        return fib(number -1) + fib (number -2);
}
```

Requirements

- Create three files
  - main() .c file
  - fib() .h file
  - fib() .c file

- Use a makefile to compile both .c files into .o files and later compile the two .o files into one executable: fib

- Example:

  $ ./fib 10

  $ 55

References

- Linux Kernel Stats: http://linuxator.wordpress.com/2008/07/22/5-things-you-didnt-know-about-linux-kernel-code-metrics/
- GNU make
www.vast.uccs.edu/~tboult/CS330/NOTES/make.ppt