Lecture 2
A Unix Primer
Topics

- Intro to unix
  - Logging in
  - Filesystems and Files
  - Useful commands
  - Editors
- Intro to make & compilers
Don’t Panic! Don’t Panic!
Don’t Panic! Don’t Panic!
Don’t Panic!
Unix

- developed in the early 1970s
- as a simpler version of MULTICS
- today freely available versions of unix-clones for all kinds of machines:
  - Linux
  - FreeBSD, openBSD, NetBSD
  - Darwin
- consider only the common concepts etc.
- remember the `man` command!
Logging in

- Each Unix machine supports an unlimited number of simultaneous users.
- You need an userid and a password to use a Unix machine.
- If you sit before a Unix machine, it’ll ask for your userid and password.
- If you want to connect to and work on a remote machine, use:
  - `ssh UID@machine.domain.name`
Shells

- unix is command driven
- graphical interfaces (X Windows) are nothing more than programs that run on a unix machine
- they are 1980’s add-ons to unix
- the command processors are called 'shells'
- they are just programs themselves
- there are a number of them: ksh, bash, csh, tcsh, sh
- they offer different add-ons to the user, e.g., command line editing and scripting
Unix uses a generalized file concept
   • ’normal’ files (tex, programs, data etc)
   • devices (disks, tapes, CDROMs, display, mice etc)
   • data & devices on remote computers

File system is hierarchical (uses sub-directories starting at a root)
Files

- file names are nearly arbitrary
- case sensitive (not always, e.g., OSX’s HFS+)
- important special characters:
  - `/`: used as directory separator
  - `*`, `?`: wildcard characters used to select groups of files
  - `\`: used to mask special characters
- tip: use alphanumeric characters for files: A–Z a–z 0–9
Canonical Files/Names

- `/dev`: directory with hardware devices
- `*.f`: fortran source code files
- `*.c`: C source code files
- `*.tex`: \TeX source code files
- tip: use extensions that follow the conventions, don't use . for something else
Essential Commands

- **man**: gets you help on a command, e.g., `man man`
- **df**: shows all file systems and their used/free space
- **pwd**: shows current directory and it’s path starting at the root
- **ls**: lists files in current directory
- **cd**: changes to a different directory
Essential commands

- **cat, more, less**: display content of a (text) file on the terminal
- **exit**: ends the shell, closes the connection to a unix machine
- **xterm**: starts a new shell in a new X Window (if applicable!)
there are many different editors available

- **vi** (pronounced as separate letters: “v i”): the standard
- **emacs**: a freely available full-screen text editor.
- **pico**: This is the editor that comes with the **pine** email program.
Programming

- ‘all’ programming languages are available on unix
- writing a program is a multi-stage process:
  - develop and write program source code (text file)
  - use a compiler to translate source to ‘object’ format
  - link object code with system libraries to produce executable program
- we will use fortran95 as programming language
Programming Support

- fortran95 compile have different names:
  - f95: most common, e.g., NAG fortran95
  - ifc: Intel fortran95 compiler for IA32
  - xlf95: IBM fortran95 compiler for OSX, AIX, Linux on PPCs

- important utility program: make
- important debuggers: gdb, dbx