

ECE 471 – Embedded Systems

Lecture 10

Vince Weaver

`http://www.eece.maine.edu/~vweaver`

`vincent.weaver@maine.edu`

3 October 2013

Announcements

- Homework #2 has been assigned. The due date is extended until the 10th.
- Midterm due date will also be pushed forward, likely to the 22nd.



Gumstix/Linux Followup

Gumstix Board costs \$200, so be careful.

Not like a Rasp-pi or STM32 Discovery



SD Card Digression

- Why so slow?
- How did I make it half the size.
- SD and flash are sort of their own embedded systems



Connecting to Gumstix with Windows

This info also available on the class website.

- Download Teraterm <http://ttssh2.sourceforge.jp/>. It's free. Get the .exe, run it, install
- Start it up. If your gumstix is plugged in you should be able to select serial/com port, and one should be listed as USB serial.
- If not, you may need to download a FTDI usb/serial driver. You can get one from <http://www.ftdichip.com>.



`com/Drivers/VCP.htm`

- You will need to go to setup/serial port menu and set the baud rate from 9600 to 115200 otherwise you will just get garbage.
- To send a file from gumstix to your machine, at prompt type `sz file_to_send`. Then go to File/Transfer/zmodem/receive and it should get it.
- To send a file from your machine to gumstix, at prompt type `rz` then File/Transfer/zmodem/send and pick the file.



- Brief history of serial ports?



Command-Line Linux

The way we did things in the old days.
Some of us still prefer the command line.

You come up in the “shell”. Default is bash, the “Bourne Again Shell” (more computer person humor). There are various shells available (bash, sh, zsh, csh, tcsh, ksh) and you can select via `chfn`.



Root Filesystem Layout

- Executables in `/bin`, `/usr/bin`
- System executables under `/sbin`, `/usr/sbin`
- Device nodes under `/dev`
- Config files under `/etc`
- Home directories under `/home`, also `/root`
- Temp Files under `/tmp`. Often wiped at reboot.



- Magic dirs under `/proc`, `/sys`
- Libraries under `/lib`, `/usr/lib`, sometimes `lib64` too
- Boot files under `/boot`
- `/usr` historically only files needed for boot in `/`, stuff that can be shared over network (or stored on a second drive if your first drive was too small) would be under `/usr`
- `/opt` often commercial software installed there



- `/srv`, `/run`, `/var` these are where server programs store data
- `/media`, `/mnt` places to mount external disks like memory keys and CD roms
- `/lost+found` where the disk checker may store lost files it finds when fixing a disk after unclean shutdown



Interesting Config Files

- `/etc/fstab` – the filesystems to mount at boot time
- `/etc/passwd` – list of all users, world readable
- `/etc/shadow` – passwords stored here for security reasons
- `/etc/hostname` – name of the machine
- `/etc/hosts` – list of local machines, usually searched before resorting to DNS lookup over network



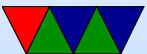
- `/etc/resolv.conf` – where your nameserver address is put
- `/etc/sudoers` – list of users allowed to use “sudo”
- `/etc/network/interfaces` – on debian the network settings are stored here



Devices

Block vs Char devices

- `/dev/sd*` – SCSI (hard disks)
- `/dev/tty*` – tty (teletype, logins, serial ports)
- `/dev/zero`
- `/dev/full`
- `/dev/random` , `/dev/urandom`



- `/dev/loop`

Network devices are an exception.



Interesting /proc Files

These files are not on disk, but “virtual” and created on-the-fly by the operating system when you request them.

- /proc/cpuinfo – info on cpu
- /proc/meminfo – memory info
- Each process (running program) has its own directory that has info about it



Processes

- Each program assigned its own number, a process id, often called a “pid”
- Can list processes with `ps -efa`
- Also can get real-time view of what’s going on in a system with `top`



Common Commands

- `ls` : list files
 - `ls -la` : list long output, show all (hidden) files. on Linux any file starting with `.` is hidden
 - `ls -la /etc` : list all in `/etc` directory
 - `ls *.gz` : show all ending in `gz`. `*` and `?` are wildcards and can be used as regular expressions.
- `cd DIR` : change directories (folders)
 - `cd ..` : go to parent directory
 - `cd .` : go to current directory



`cd /` : go to root directory

`cd ~` : go to home directory

- `cat FILE` – dump file to screen (original used to conCATenate files together but more commonly used to list files)
- `more` / `less` – list contents of file but lets you scroll through them. `less` more advanced version of `more`
- `exit` / `logout` / `control-D` – log out of the machine
- `df` / `du` – show disk space



`df -h` pretty-prints it

- `man command` – show documentation (manual) for a command. For example `man ls`
- `rm` remove file. CAREFUL! Especially famous `rm -rf`. In general on Linux you cannot undo a remove.
- `cp` copy file. CAREFUL! By default will overwrite the destination without prompting you.
- `mv` move file. CAREFUL! Can overwrite!
`mv -i` will prompt before overwrite



- `tar` create archive file `tar cvf output.tar dir`
`tar xzvf output.tar.gz` uncompresses a `.tar.gz` file
- `gzip` / `gunzip` / `bzip2` / `bunzip2` compress/uncompress a file. `gzip` and `bzip2` are two common formats, many more exist



Compiler / Devel Commands

- `make` – build a file based on list of dependencies in Makefile
- `gcc` – C compiler. Simplest something like this: `gcc -O2 -Wall -o hello hello.c`
- `g++` C++ `gfortran` Fortran
- `as`, `ld` – assembler and linker
- `gdb` – debugger



- `strace` – list system calls
- `git` – source code management



Other Commands

- `shutdown` – used to shutdown / reboot
- `last` – list last people to log in
- `su` / `sudo` – switch to root, run command as root
- `uptime` – how long machine has been up
- `date` – show the date
as root you can use `date -s` to set the date



- `whoami` – who are you
- `write` / `wall` / `talk` – write to other users
- `finger` – get info on other users
- `w` / `who` – see who is logged in
- `wc` – count words/bytes/lines in a file
- `dmesg` – print system and boot messages
- `ln` – link files together, sort of like a shortcut



`ln -s goodbye.c hello.c` – symbolic link. also hard links

- `dd` – move disk blocks around, often used for creating disk images
- `mount / umount` – mount or unmount filesystems
- `mkfs.ext3` – make new filesystem
- `e2fsck` – filesystem check
- `ifconfig / route` – show and setup network config



- `dpkg / apt-get update/upgrade/install` – debian only package management
- `ssh / scp` – log into other machines, copy files remotely
- `lynx` – text-based web browser
- `reset` – clear the screen and reset settings (useful if you accidentally cat a binary file and end up with a screenful of garbage). Control-L also refreshes the screen
- `linux_logo` – my program



Editing files

Linux and UNIX have many, many editors available. Most famous are vi and emacs. On our board using nano might be easiest.

- nano – a simple text editor.

`nano FILENAME` – edit a filename

It shows the commands you can do at the bottom. `^O`

means press control-O

control-O : writes

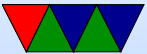
control-X : exits



control-W : searches

control-\ : search and replace

control-C : prints line number



Redirection and Pipes

- redirect to a file : `ls > output`
- redirect from a file : `wc < output`
- pipe from one command to another : `ls | wc, dmesg | less`
- re-direct stderr : `strace 2> output`



Suspend/Resume

- Press control-C to kill a job
- Press control-Z to suspend a job
- Type `bg` to continue it in the background
- Type `fg` to resume it (bring to foreground)
- Run with `&` to put in background to start with. (ie, `mpg123 music.mp3 &`).



Permissions

- user, group – use chgrp
- read/write/execute – use chmod



Shell Scripts

- Create a list of files in a dir
- Start with the shell, `#!/bin/sh` (or perl, etc)
- Make executable `chmod +x myfile`



Command Line History

- Can press “tab” to auto-complete a command
- Can press “up arrow” to re-use previous commands
- Can use “control-R” to search for previous commands



Environment Variables

- `env`
- Varies from shell to shell.
- `export TERM=vt102`
- `PATH`, and why “.” isn’t in it. This is why you have to run self-compiled binaries as `./blah`

