

Driver Basics



Getting our hands dirty
ECE 373

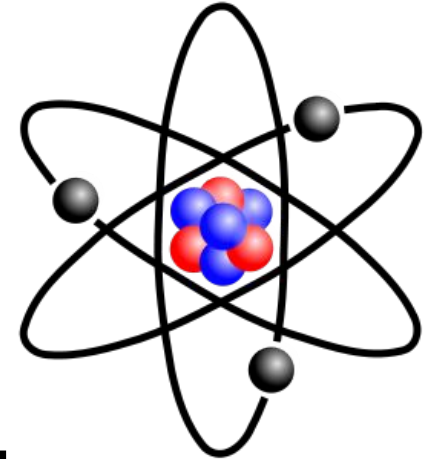
Prelims

- Questions on class?



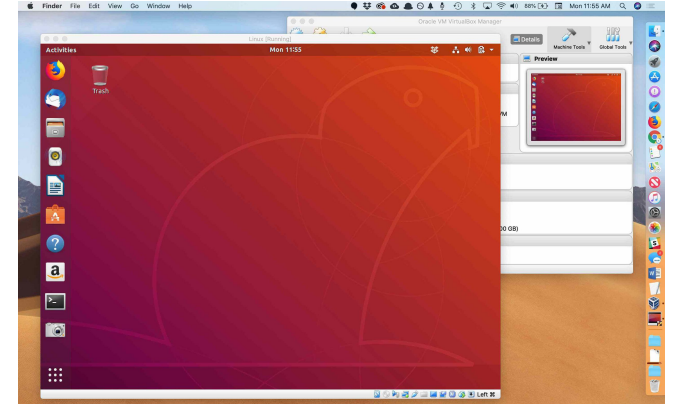
Fun with Atoms and VM's

- Serial link to Atom box
 - Null modem serial cable
 - Putty and minicom
- Screen capture and session logging
 - Putty and minicom logs
 - Script command

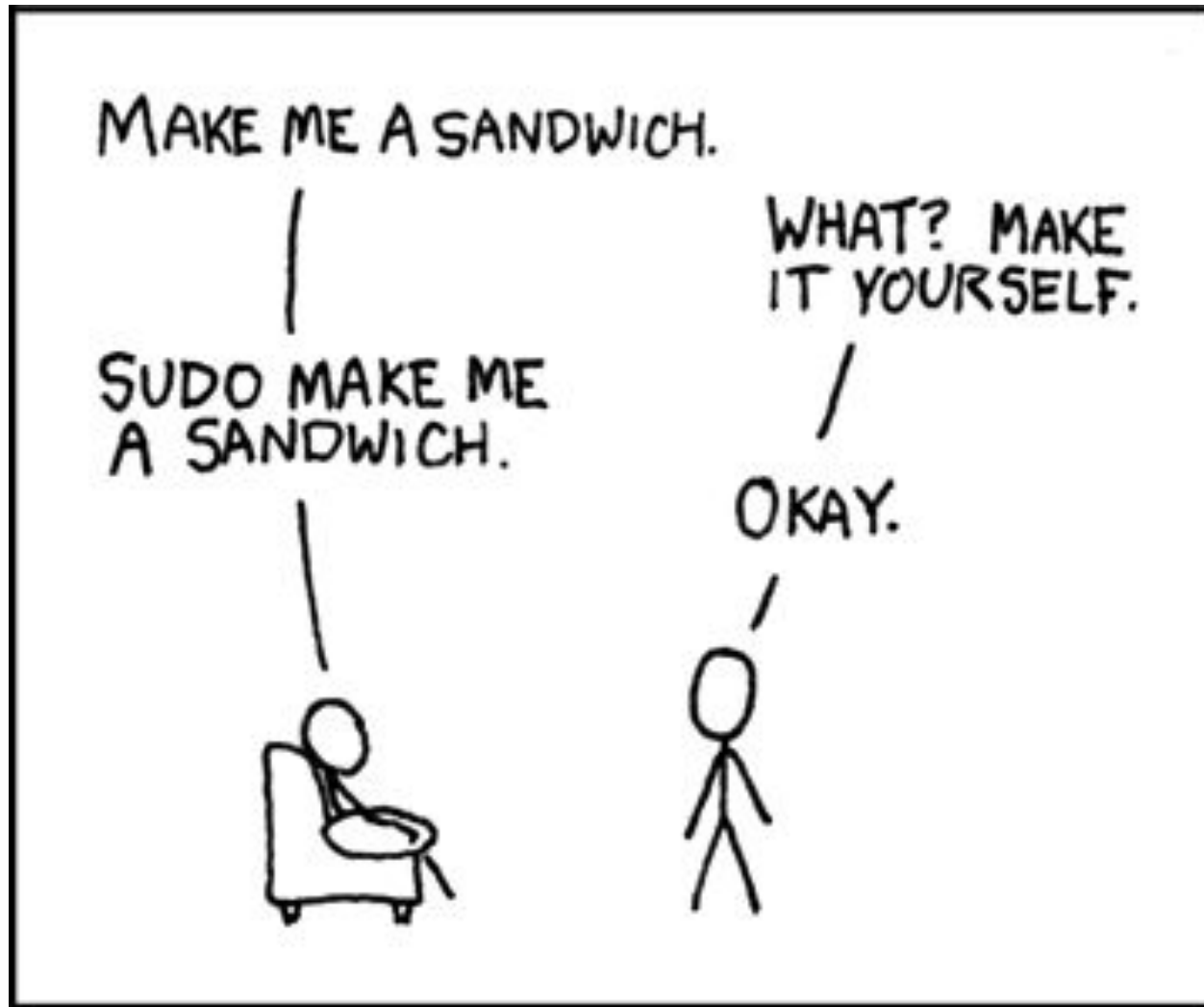


Virtual Machines

- "Serial" link to machine
 - Connects to a COM port
 - Also putty and minicom
- Also screen capture and session logging
- Emulates full computer inside of a "bare metal" machine
- Can load different "hardware" to emulate behavior
- Can direct-assign hardware into a VM as well



Running as root



<http://xkcd.com/149/>

File Sharing – USB memory

- GUI goodness

- Ubuntu GUI automounts USB
- `/media/<volume-name>`

- Do-it-yourself

- `mkdir /media/usb`
- Plug in USB drive
- **Look for** `/dev/sdX` **in** `/var/log/messages` **or** `syslog`
- `mount /dev/sdX /media/usb`
- `umount /dev/sdX`



Virtual Machines

- Hypervisor (thing that runs VM) integrates with host
 - Hypervisor examples:
 - KVM
 - Hyper-V
 - VMware and friends
 - VirtualBox
 - Parallels (Mac-specific)
- Host == “bare metal”
- Share files between VM and host
 - Hypervisors have shared folders to map
 - ssh into VM (may need to install openssh-server)
 - Virtual “USB stick” for VM

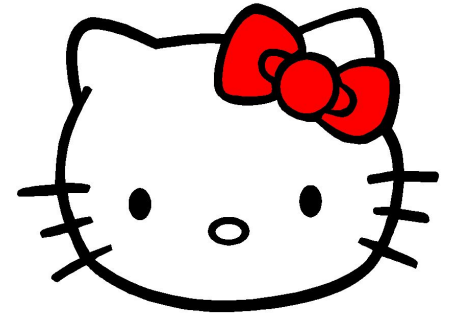
Hello Kernel

```
#include <linux/init.h>
#include <linux/module.h>
MODULE_LICENSE("Dual BSD/GPL");

static int __init hello_init(void)
{
    printk(KERN_INFO "Hello, kernel\n");
    return 0;
}

static void __exit hello_exit(void)
{
    printk(KERN_INFO "Goodbye, kernel\n");
}

module_init(hello_init);
module_exit(hello_exit);
```



Plug-In APIs

- Application Programming Interface
- Defines "hooks" on which to hang your driver
- Standard programming connections



Bits and pieces

- Standard API for all driver modules
- Minimal callback hooks and Compile headers
 - Basic `#include's`
 - `MODULE_LICENSE(lic)` – legal strings
 - `__init, __exit`
 - `module_init(func)`
 - `module_exit(func)`
- `printk()` **and** `KERN_XXX`
- Kernel Makefile convolutions
- Compile command

Makefile

Makefile

```
KERNEL_DIR = /lib/modules/$(shell uname -r)/build  
PWD := $(shell pwd)
```

```
obj-m += hello_kernel.o
```

```
default:
```

```
    $(MAKE) -C $(KERNEL_DIR) M=$(PWD) modules
```

```
clean:
```

```
    $(MAKE) -C $(KERNEL_DIR) M=$(PWD) clean
```



Tabs, not spaces

Try it out

- **Compile:** `make`
- **Load:** `insmod hello_kernel.ko`
- **Results**
 - `lsmod`
 - `dmesg` – more detailed, but limited circular buffer
 - `tail /var/log/syslog` – managed syslog text file, not as much detail
 - `journalctl` – for your systemd overlords
- **Remove:** `rmmod hello_kernel`

