LTSP

Joey Fisher & Lachele Foley
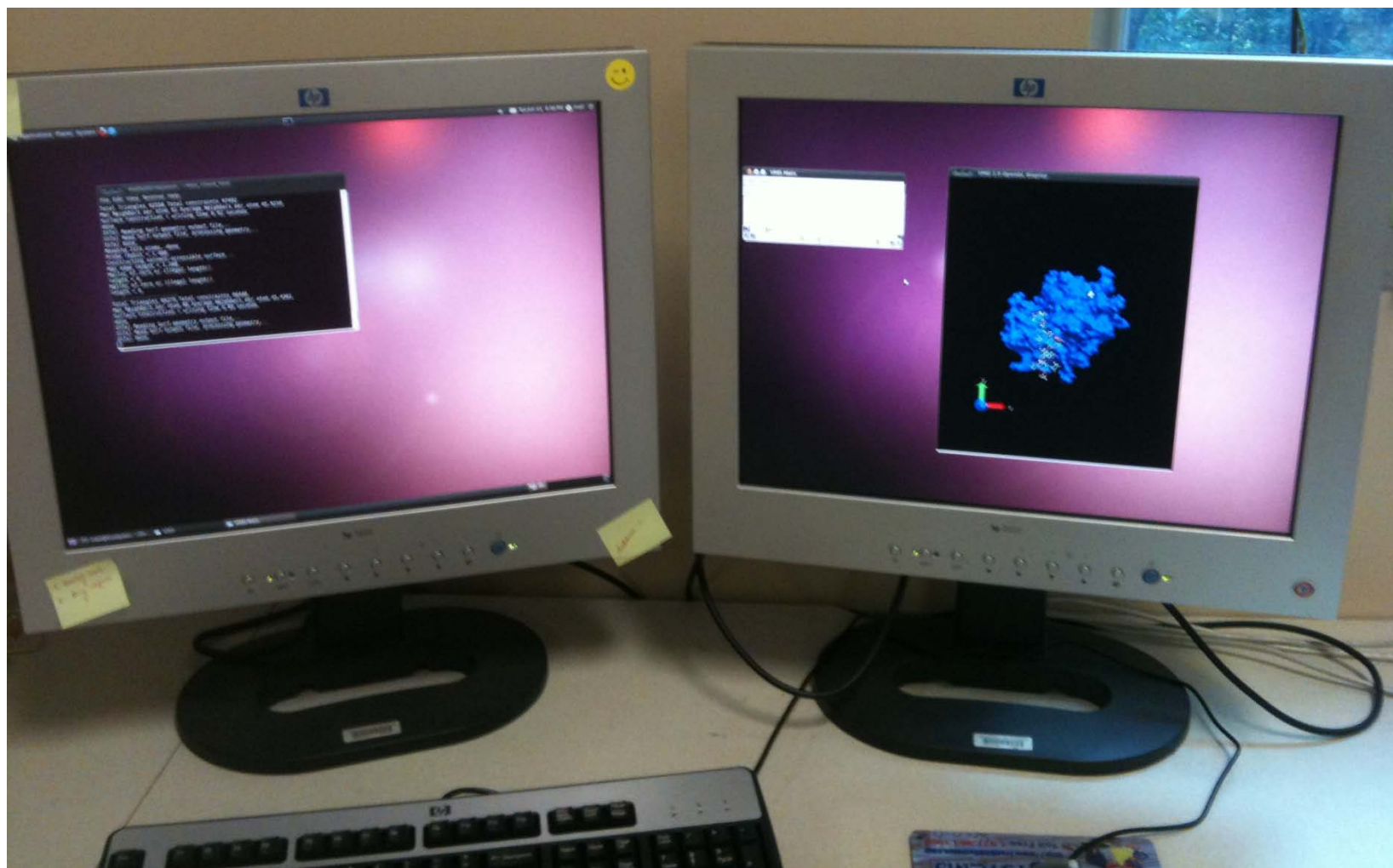
CHUGALUG 20120315

NOT YER OLDE TYME TERMINALS

"Terminal" used to be this:
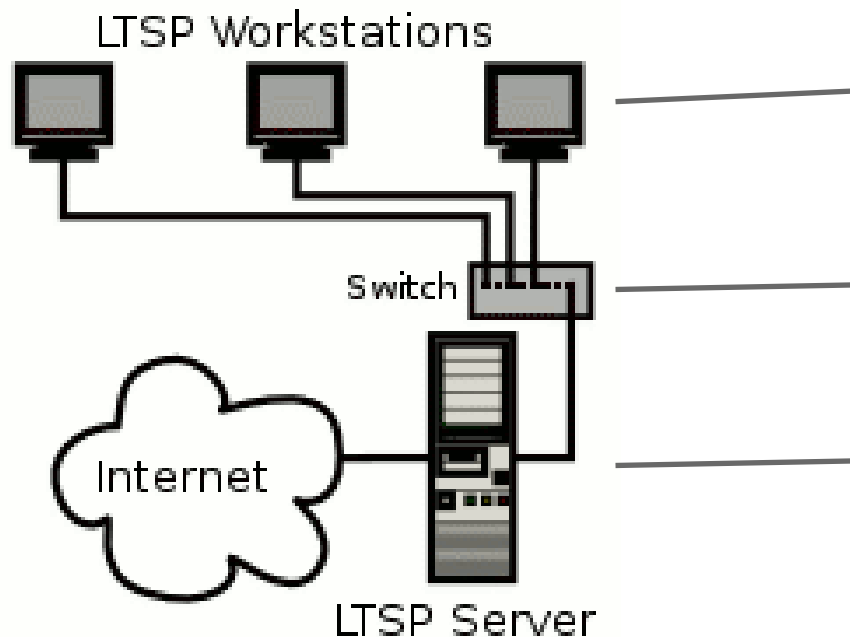
But now, it can be this:

THE BASICS

LTSP

- Linux Terminal Server Project
  - Remote computers as terminals of a main server

LTSP Workstations

Keep the old, slow computers
Might need new video, NIC
Or, get fast computers

Switch

Get a switch per your needs

Internet

Buy one beefy server

LTSP Server

http://ltsp.org/images/ltsp_diagram.gif

How, briefly

- Server network-boots the client
- Boot image has a login screen (over ssh)
- After login
  - Graphical session starts on server
  - Display is pushed to client
  - All I/O goes to/from server, as if logged in directly
    - Mostly, that is.
    - This behavior can be altered somewhat

WHY?

Big picture reasons

- Cost
  - One server is cheaper than many workstations
- Admin simplicity
  - Manage server and one image
- Community
  - Everyone is on the same machine = easy sharing
- Resource sharing
  - Productivity bursts, deadline crunches

Configurable

- Can be locked down
  - K-12 school setting, public kiosk
- Can be set free
  - Office or research workgroup
- Can run on limited resources
  - Slower, of course, but quite usable
- Can be screamingly fast
  - Well... mostly... see later

Runs on several systems

- Buy whole solution if you like:
  - http://www.disklessworkstations.com
  - Commercial variants also available
- Common Linuxes supported
  - Requires tight integration with OS
  - I have tried Debian and Ubuntu
    - Ubuntu is really easy, but Debian isn't bad at all

JUST BECAUSE I CAN

A demonstration

This laptop

Server over there

Base OS: Win XP

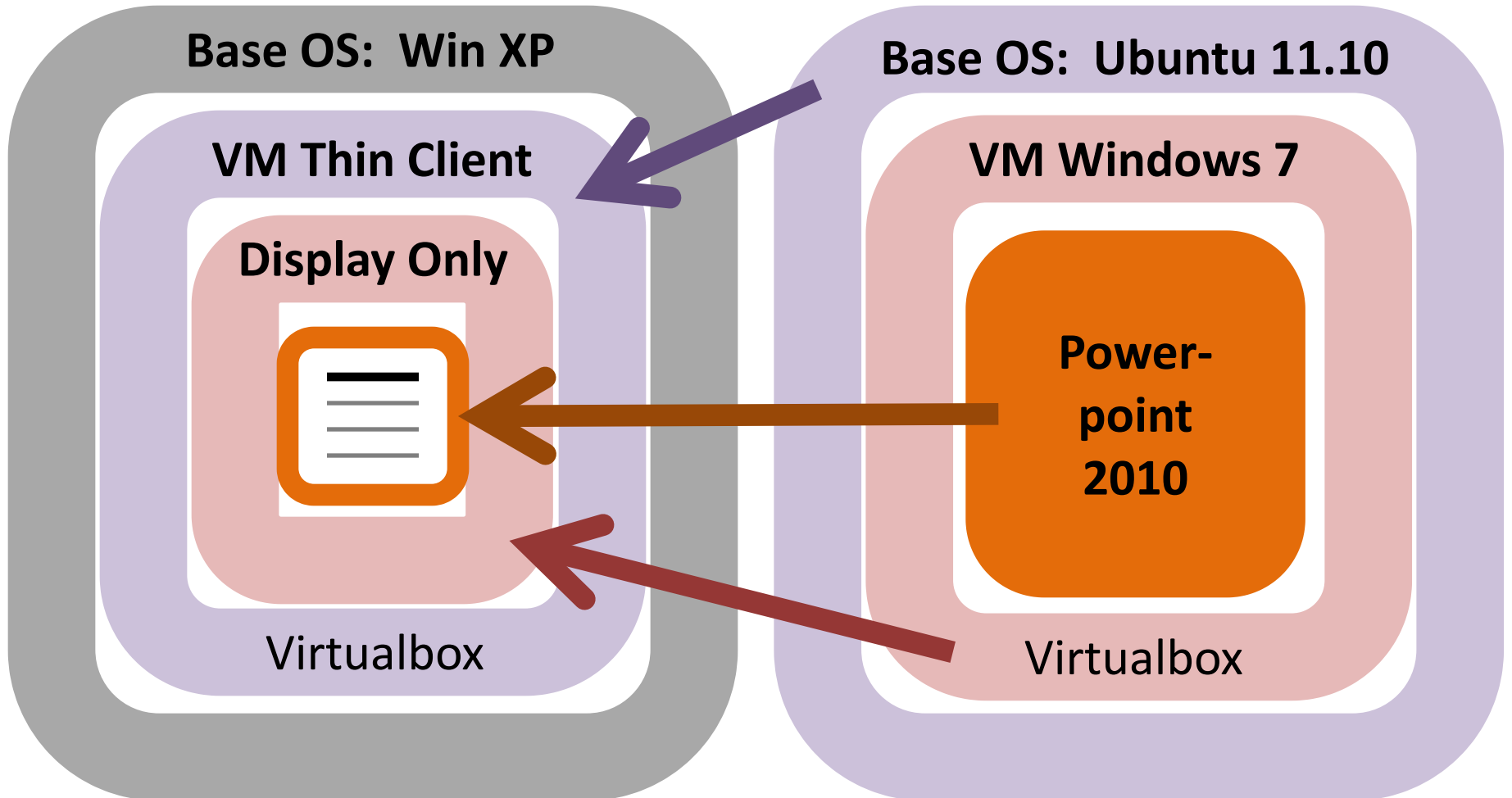**Base OS: Ubuntu 11.10**

VM Thin Client

VM Windows 7

Display Only

Power-point 2010

Virtualbox

Virtualbox

CONFIGURATION NOTES

Server

- CPU's and memory
  - Depends greatly on user requirements
    - Running VM's? Fancy software?
- The demo machine I brought has:
  - Quad-core Intel Core-I7 3.40 GHz
    - Hyperthreaded so 8 procs
  - 4 GB memory
  - GigE NIC for clients and internet
    - Two NIC's better for most setups

Clients

- Need NIC that will PXE boot – or –
  - Can use CDROM to boot from non-bootable NIC
    - Etherboot Project: http://www.rom-o-matic.net/
  - Might need to enable PXE boot in bios
- Performance affected by:
  - CPU speed, cache, etc.
  - Local memory
  - Video card
  - NIC – all activity goes over the NIC

More on clients

- The server can only do so much
- For my P4+1GB at home:
  - Native online video is choppy (good card)
    - Ok for small, but full-screen is bad
  - Server doesn't improve, actually degrades a bit
    - Sends too much info, maybe? Not sure why.
- For very good video, in my experience, you need good client hardware
  - Video cards on client and server matter, too

NIC & Switch

- ## 10/100
  - Ok for smaller displays; video can be choppy
  - The demo was over a 10/100 NIC
  - Still can be better than the old computers alone
- ## GigE & better
  - Needed for high performance
  - Difference obvious even on old computers
  - Haven't tried 10GigE, fiber, infiniband, etc.
    - Might have hardware issues, might not

Video

- Most video happens on the server
  - Can help to have a good video card there
- The division isn't so perfect, though
  - Some software integrates closely with video card
  - Example: VirtualBox
    - Server and client, hardware and software
- Proprietary drivers
  - Hard/impossible to install into client chroot
  - Can overwrite library files (bad for heterogeneous)

VIDEO...

Can have video headaches

- Very new or very old hardware
  - Couldn't even install 11.04 directly on some new
  - But, did manage to boot as client (much trial)
- Use other than the standard session
  - For example, XFCE rather than Gnome
- Heterogeneous hardware
- Client hardware different from server
- But, can usually make work

Really… I mean headaches


Possible causes of this (seen often):
- Boot 64-bit client with 32-bit image
- Bad response of non-NVIDIA card to NVIDIA libs
- On upgrade from Maveric to Natty
  - AMD server, on-board graphics (no NVIDIA)
  - Happened on some ATI/AMD machines
  - But, also old on-board Dell graphics

Other video headaches

- Sudden boot back to login
  - Not caused by too high an XRAMPERC
- Cryptic error messages
  - Might reference server or client
    - Hard to tell which one is missing what
  - App might not run at all
- Default Ubuntu (3D) session
  - Might not work, even with good card
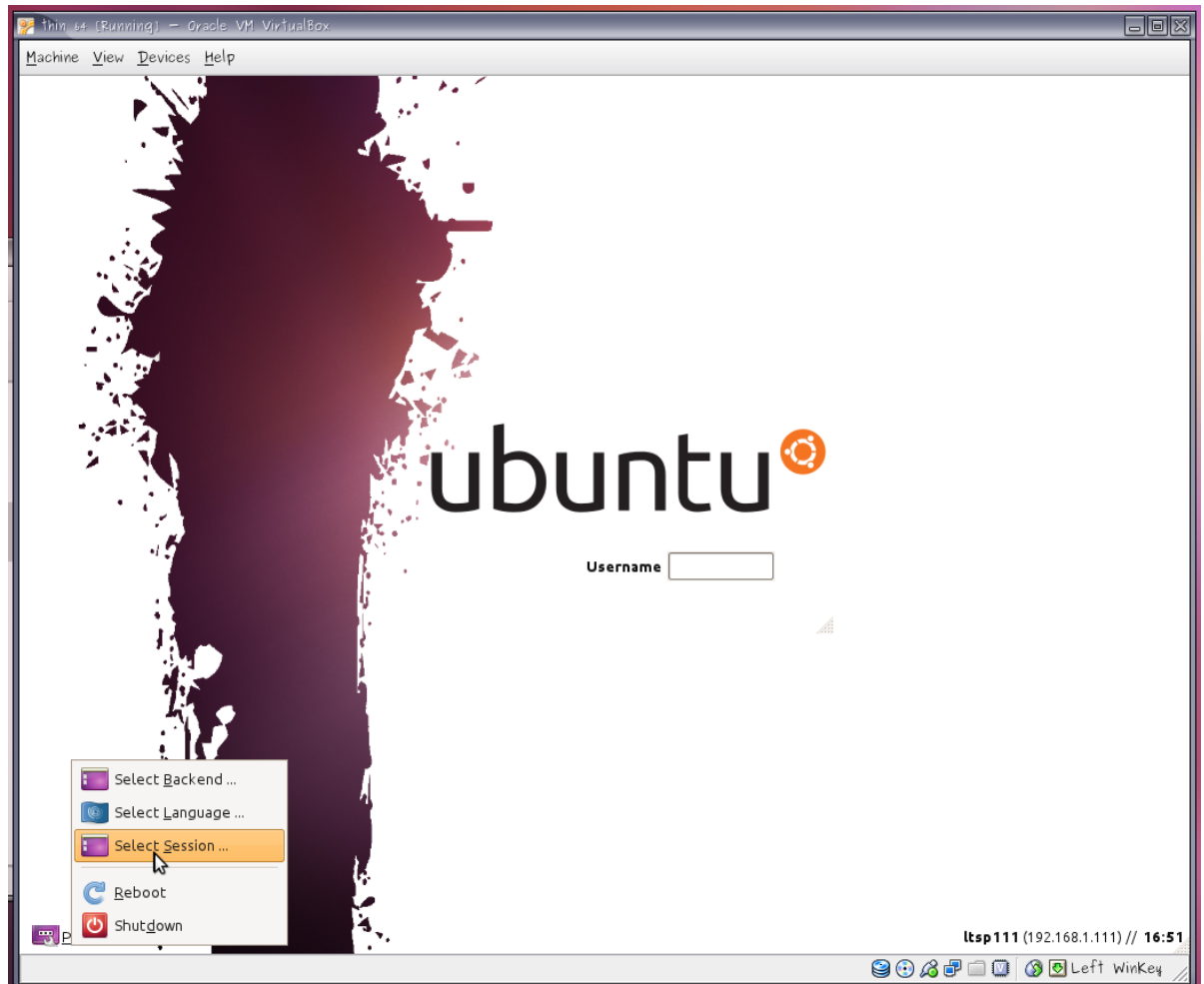   - Lots of people hate it anyway

When all else fails: failsafe xterm

Standard login screen – this isn't using much of the client video yet.
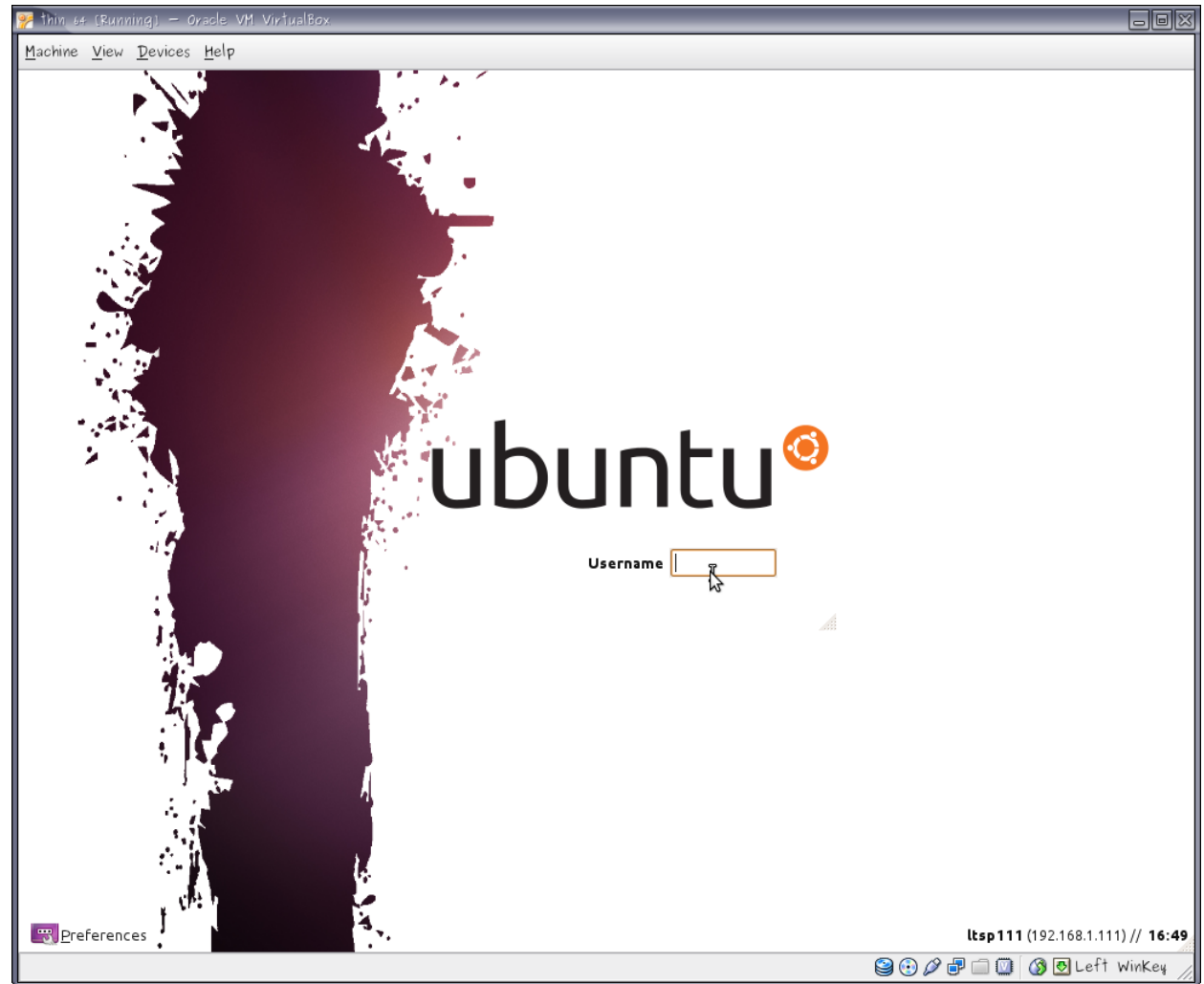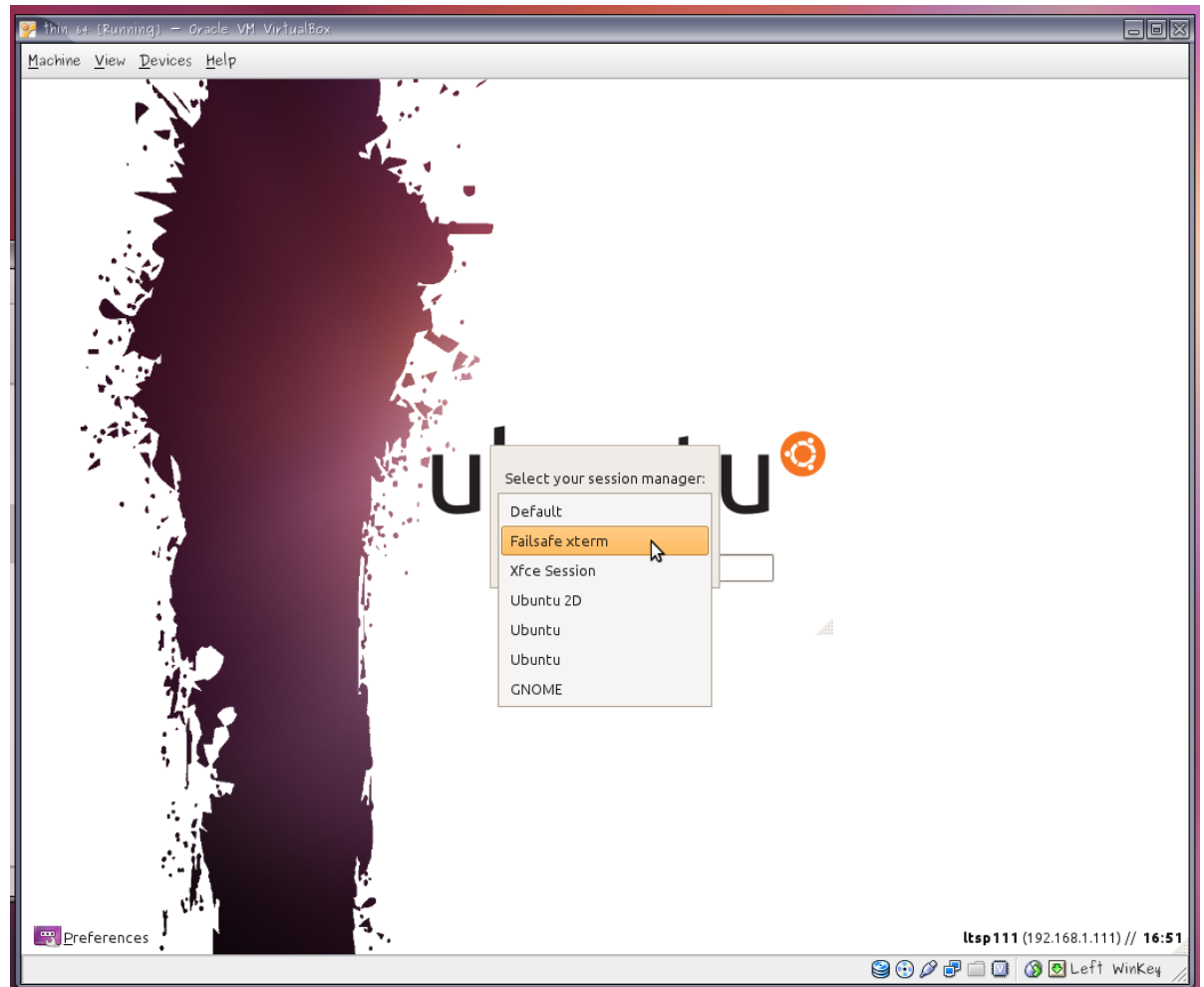
See options at lower left

To get a plain command-line prompt, choose select session

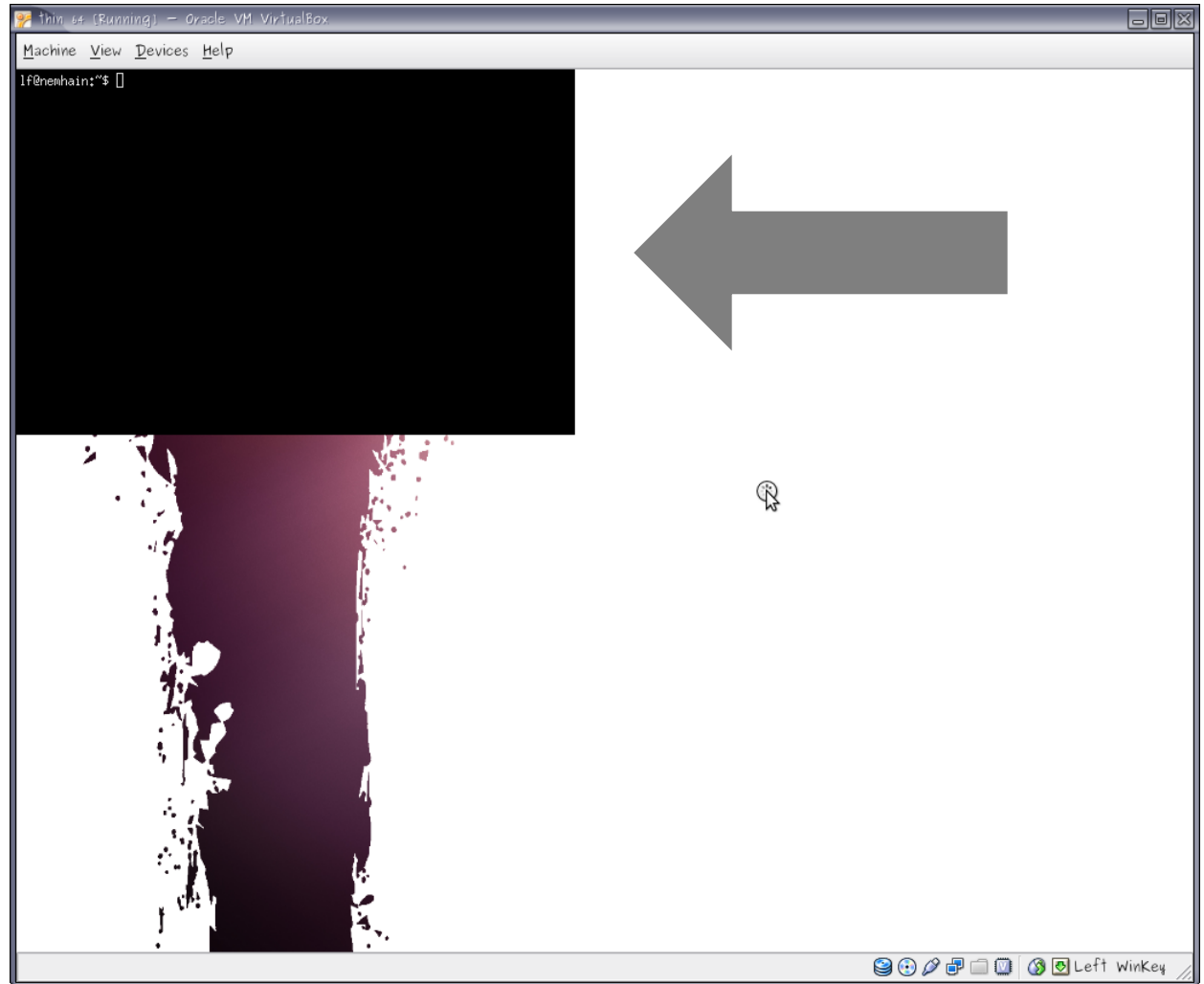# When it just won't work

Then choose
Failsafe Xterm

Explore other
options once
the video works

This is the Xterm

Make an xorg.conf file for the local hardware (save to a nonchroot location). Add to chroot later.

(They say not needed in /opt/ltsp/<arch> and sometimes that seems correct.)

LITTLE THINGS

USB devices

- Most USB drives/media recognized
- USB on 64-bit client won't work
  - Weird behavior
  - New version out soon – hope to be fixed
- Most USB printers should work
  - Mine won't play with Linux at all
  - Must be on server for VirtualBox to use
- Cameras, special items: hit or miss

Other devices

- Most CD/DVD drives/media recognized
- I haven't made a movie DVD work on client
  - Sees DVD – software doesn't cooperate
- Some formats/devices usable only on server
  - Can use from a client; must plug in to the server
  - For example: camera or some MS install disks
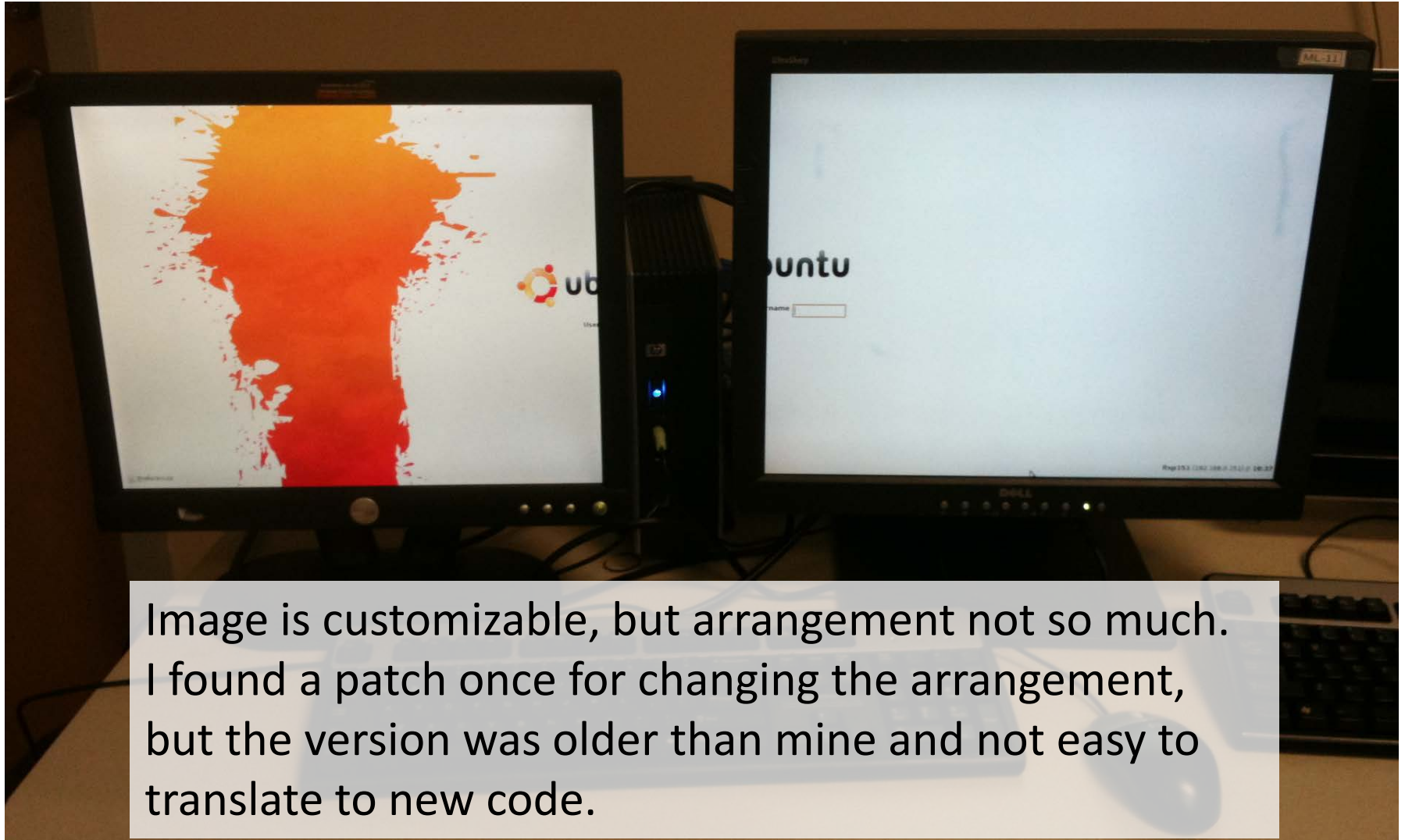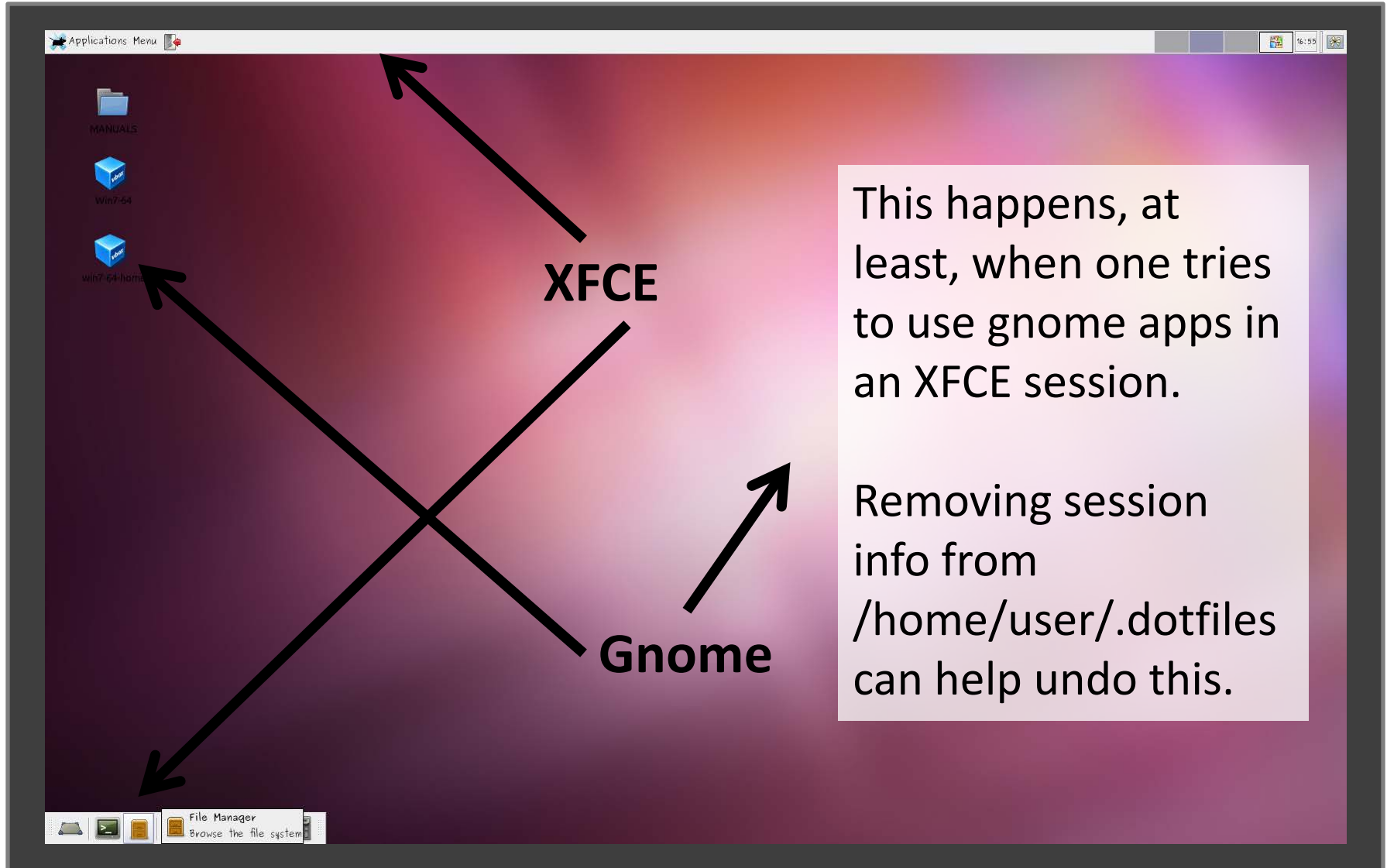- Haven't made client parallel port work yet ☺

Split login screen

Image is customizable, but arrangement not so much. I found a patch once for changing the arrangement, but the version was older than mine and not easy to translate to new code.

Gnome can be overly assertive

My normal desktop
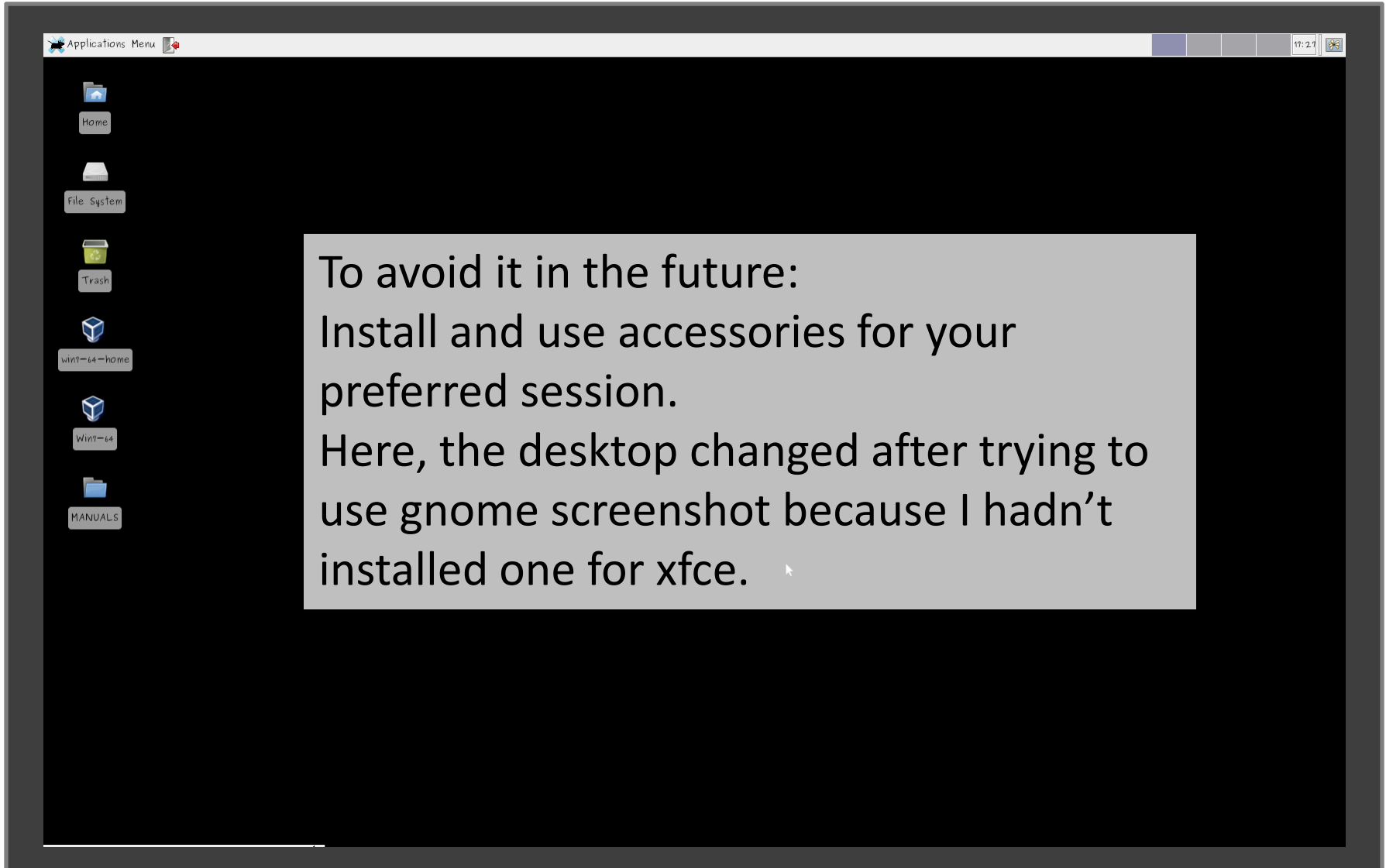
SHALL WE BREAK?

New version coming soon, might differ

HOW TO DO IT (UBUNTU)

Installing LTSP on Ubuntu

- Very easy
  - Just follow instructions:
    https://help.ubuntu.com/community/UbuntuLTSP/LTSPQuickInstall
- Need the Alternate CD
  - Can take some looking – not just the "server" CD
- If not on an isolated network:
  - Segregate from other networks after install
    - That is, unplug the cable
  - Ensure built-in DHCP doesn't clobber

After installation

- Set up the dhcpd.conf file
  - In /etc/ltsp/dhcpd.conf

- Build new images as needed
  - Likely need 32-bit image for now
    - Recall 64-bit USB issue

- Unless you have something fancy, that's it
  - An example of fancy is very old or new hardware

Setting up DHCP

- Default
  - Only LTSP on your local network
  - Only one type of client, same arch as server
- Must config for:
  - Alternate or mixed architectures
  - Multi-purpose local network

Non-default DHCP

- I uninstall NetworkManager (YMMV)
- Might need more than one NIC
  - For multiple users
  - Can load-balance or split users among them
- As an aside…
  - Note that the client's network is only LTSP
    - No internet access
    - This can be changed

/etc/network/interfaces

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.1.46
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
    broadcast 192.168.1.255

# LTSP dhcp interface
auto eth1
iface eth1 inet static
    address 192.168.21.254
    netmask 255.255.255.0
    network 192.168.21.0
    broadcast 192.168.21.255
    mtu 9000
```

This setup causes "martian source" errors in the log files. Those probably go away with a different netmask.

Haven't tried for fear what else will break if I do that, and don't want them talking to each other anyway, not that it would really matter.

Any folks here who know?

```
not authoritative;                              /etc/ltsp/dhcpd.conf
deny unknown-clients;


subnet 192.168.21.0 netmask 255.255.255.0 {
 INTERFACES="eth1";
 pool { #
      range 192.168.21.1 192.168.21.10;
       filename "/ltsp/i386/pxelinux.0";
       option root-path "/opt/ltsp/i386";
       option subnet-mask 255.255.255.0;
      host P4_1 { # some P4 machine
        fixed-address 192.168.21.1;
        hardware ethernet 01:23:45:67:89:AB;}
    }
 pool { #
      range 192.168.21.11 192.168.21.20;
       filename "/ltsp/speciali386/pxelinux.0";
       option root-path "/opt/ltsp/speciali386";
       option subnet-mask 255.255.255.0;
      host SpecialNeeds_1 { # some other 32-bit machine
        fixed-address 192.168.21.11;
        hardware ethernet 02:23:45:67:89:AB;}
    }
 }
```

Chroot

- Client chroot is read-only to the client
  - This frustrates certain installs
- Building the image currently uses all procs
  - New version should fix that
- Generally need to rebuild image after
  - Upgrading client chroot
  - Upgrading server
  - Changing/installing authentication software

Build client

- Can customize, but default is simple
  - `ltsp-build-client --arch <arch> --chroot chrootname`
    - `<arch>` = i386 and amd64 and maybe others
    - Use "chroot" to have mutliple variants
      - E.g. simplei386, kioski386, nvidiaamd64
  - `ltsp-update-sshkeys`
  - `ltsp-update-kernels`
- Don't forget to update dhcpd.conf

Update client

- Can apt-get update/upgrade
  - Instructions online
  - Can install software into the client chroot
    - Doesn't bloat boot image
- Simple enough:
  - `ltsp-update-image --arch` **`chrootname`**
- Don't always need to update keys/kernels
  - But, doesn't hurt and is fast

Client configuration

- Can customize a lot
- /var/lib/tftpboot/<arch>/lts.conf
  - Localdev
    - Make local devices visible
  - Localapps
    - Allow running apps on the client hardware
      - Not very useful if your clients are slow
  - Set xorg.conf and other things
- Manual has many options

Some important locations

- /opt/ltsp
  - Location of client chroots and images
- /var/lib/tftpboot/ltsp
  - Location of boot image and some config files
    - Notably lts.conf (not present by default)
  - /var/lib/tftpboot/ltsp/<arch>/pxelinux.cfg/default
    - Delete "quiet" and "splash" if trouble with booting
- /etc/ltsp/dhcpd.conf

LTSP Kiosks

Lock'em down &

Keep'em clean

To Authenticate or Not?

- First, you must decide whether to allow unauthenticated users access or not

- This is largely determined by your corporate policy

- UGA has a standing exemption for kiosk-type appliances, so long as all unnecessary services and applications are inaccessible

Authenticate Against Directory

- Should you choose, it's ideal to authenticate users against an Active Directory

- sudo apt-get install likewise-open

- Sudo domianjoin-cli join <yourdomain.local> <admin-domain-acct>

- More on this later

Straight to Desktop (No Auth)

- You must create a local user account to be the kiosk user

- You can specify the logon user in lts.conf:

- ```
  [Default]
  LDM_AUTOLOGIN = True

  [192.168.1.101]
  LDM_USERNAME = user1
  LDM_PASSWORD = password1

  [192.168.1.102]
  LDM_USERNAME = user2
  LDM_PASSWORD = password2
  ```

Locking Down the Interface

- Need to use a lightweight and highly customizable window manager

- Icewm is my personal choice as it is extremely lightweight and customized by a handful of config files
  - Sudo apt-get install icewm

- Modify lts.conf so that icewm is default wm

- ```
  [Default]
  LDM_Session = icewm
  ```

IceWM

- 5 config files to worry about
  - Keys, preferences, startup, toolbar, winoptions
- Usually installs to /usr/local/share/icewm
- Preferences is the biggest file that gives all the UI options
- Toolbar controls the "start" menu, can be stripped down to just show a single icon

Security

- Password BIOS & disable boot to CD/USB
- Kiosks will trust web sites more than the user
- http://ikat.ha.cked.net
- Disable xterm easily by chmod –x xterm

Firefox

- Several "kiosk mode" plugin extensions to choose from
  - I prefer KioskFox
- Extensions work on a per-user basis, so this works best with the auto-login users
- Launch Firefox at start by editing the startup file in icewm dir

Gathering Usage Metrics

- For the auto-login kiosks, we can assume that every time firefox is launched, it is a new user

- Let's make a script that launches firefox, and call that script at startup instead of directly calling firefox, so we can do more than just launch the browser

- ```
  !/usr/bin/perl
  my $status = system("/usr/bin/firefox http://my.uga.edu &");
  my $status = system("/usr/local/bin/counter.sh");
  ```

Daily Count

- Counter.sh will simply:
  ```
  echo 1 >> /usr/local/bin/count
  ```
- Next we'll create a count-daily.sh script and cronjob it to run at 11:58pm:
- ```
  #!/bin/bash
  NOW=$(date +"%b")
  var=$( wc -l /usr/local/bin/count | cut -d" " -f1)
  echo $var >> /usr/local/bin/counters/log-$NOW.log
  rm /usr/local/bin/count
  touch /usr/local/bin/count
  chmod a+w /usr/local/bin/count
  sendEmail -f sender@server.com -t
  destination@server.com -u "Daily Kiosk Count: $var"
  -m "$var" -s yourmail.server.com
  ```

Monthly Count

- Similar to daily, create a new count-monthly.sh and cronjob it for 11:59pm on the last day of the month:

- ```
  #!/bin/bash
  NOW=$(date +"%b")
  awk '{tot=tot+$1} END {print tot}'
  /usr/local/bin/counters/log-$NOW.log >
  /usr/local/bin/counters/$NOW-sum.log
  Month=$(awk '{tot=tot+$1} END {print tot}'
  /usr/local/bin/counters/log-$NOW.log)
  sendEmail -f sender@server.com -t
  destination@server.com -u "$NOW Kiosk Users: $Month" -m
  "$Month" -s mail.server.com
  ```

Last Day of the Month?

- Cron can't easily tell if it's the last day of the month
  - So, the cronjob actually runs *every* day
  - And calls a script first to see if it's the last day:
  -
    ```
    #!/bin/bash
    TODAY=`/bin/date +%d`
    TOMORROW=`/bin/date +%d -d "1 day"`
    # See if tomorrow's day is less than today's
    if [ $TOMORROW -lt $TODAY ]; then
            exit 0
    fi
    exit 1
    ```
  - If it really is the last day of the month, then it should proceed with the monthly shell script
- You can use this for anything else you need to run on the last day of the month!

Log Maintenance

- It's a good idea to crunch and kill those log files after a while so that 10 years from now the poor sysadmin who inherits your kiosk setup doesn't unexpectedly encounter a full drive

- Create another shell script and cronjob it every month or so

- ```
#!/bin/bash
find /usr/local/bin/counters -mtime +180 -type f \(! -iname "*.gz" \) -exec gzip {} > logger \;
find /usr/local/bin/counters -mtime +365 -exec rm {} > logger \;
```

Kill the Idlers

- People will walk away from the terminal leaving their bank web site open
  - (I've seen this)
- I set a 3 minute timeout to activate the screensaver
- When the screensaver activates, it kills the session
- This is accomplished by a perl script that's called at startup

Watch & Kill Xscreensaver

- The perl script:

- ```
  #!/usr/bin/perl
  open (IN, "xscreensaver-command -watch |");
  while (<IN>) {
  if (m/^(BLANK|LOCK)/) {
  system "killall5";
  }
  }
  ```

- When the screensaver is deactivated, the auto user login will engage, and your startup scripts will run launching firefox

But what about those users logging in?

- They all get a home directories
  - That can grow very quickly
- To clean up, cronjob the following script:
-
```
#!/bin/bash
for DIR in `find /home/local/MYID/ -maxdepth 2 -name
'.gvfs'`
do
  USER=$(basename `dirname ${DIR}`)
  last -1 -R ${USER} |grep ${USER} |grep "still logged
in" > /dev/null
  LOGGED_IN=$?
  if [ ${LOGGED_IN} == 0 ]
  then
    :
  else
    umount -fl ${DIR}
    rm -rf `dirname ${DIR}`
  fi
done
```

LTSP Unchained

Workgroup in a secure network?

Set your users free

User freedom and speed

- Make X not go over ssh for main session
  - `LDM_DIRECTX=True`
- Unlock root in the chroot
  - Allows users to mount/use unsupported devices
    - Search for "ltsp unlock chroot root"
- Allow installations on server
  - Be liberal with sudo if appropriate
  - Consider an "installer" account
    - Let them (more) safely install 3rd party apps at will
- Redirect reboot and shutdown!

User freedom and speed

- Change home directories to world-read
  - Also new file creation defaults
- Let them see each others' files when local
  - Client chroot only mounts logged in user space
  - Can get around this
    - `LOCAL_APPS_EXTRAMOUNTS=/user_common`
    - And set sym links to desired shared data
- Set a work-ready default session
  - `LDM_SESSION=xfce4-session  # I like this one`