Introduction to Linux System Administration for the HP-UX Administrator

Rob Lucke

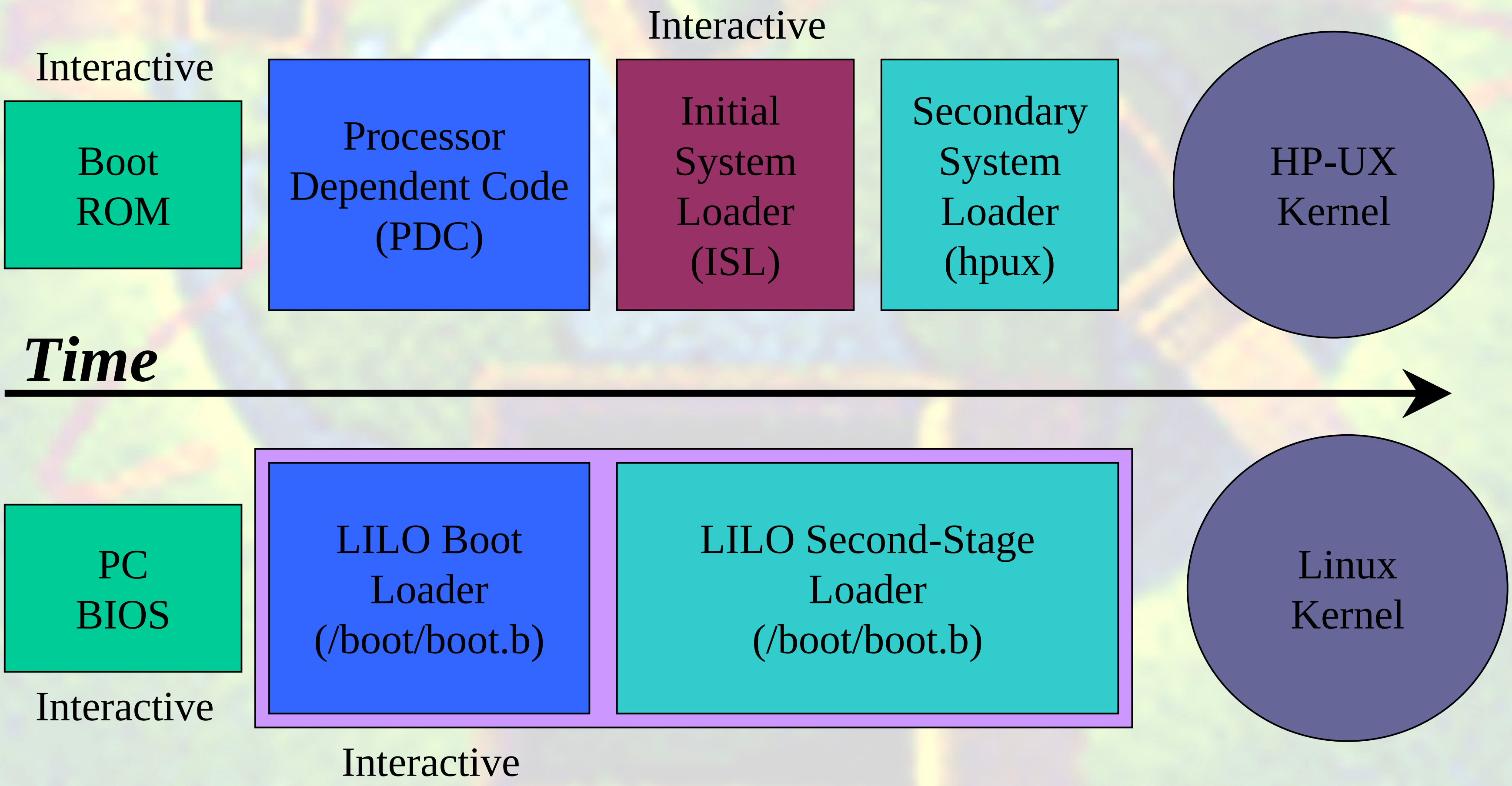
Hewlett-Packard
Americas Technical Computing
Advanced Technology Center (ATC)^2

Rob_Lucke@hp.com

Introduction to Linux

- Linux is an *open-source* operating system that comes in many different flavors, or distributions -- this seminar is based on information from the Redhat Linux version 6.1 distribution
- Linux is Unix, like HP-UX, but has differences that we need to consider as HP-UX system administrators
- This is not an exhaustive tutorial, just some of the things that I have run into during my exposure to Linux, aimed at the Linux curious, *not experienced Linux users or administrators*
- We will jump right to system administration topics, excluding the actual system installation procedure

Comparing the HP-UX and Linux Boot Processes



Interactive

Boot ROM

Processor Dependent Code (PDC)

Interactive

Initial System Loader (ISL)

Secondary System Loader (hpux)

HP-UX Kernel

Time

PC BIOS

Interactive

LILO Boot Loader (/boot/boot.b)

LILO Second-Stage Loader (/boot/boot.b)

Interactive

Linux Kernel

The Linux Loader: LILO

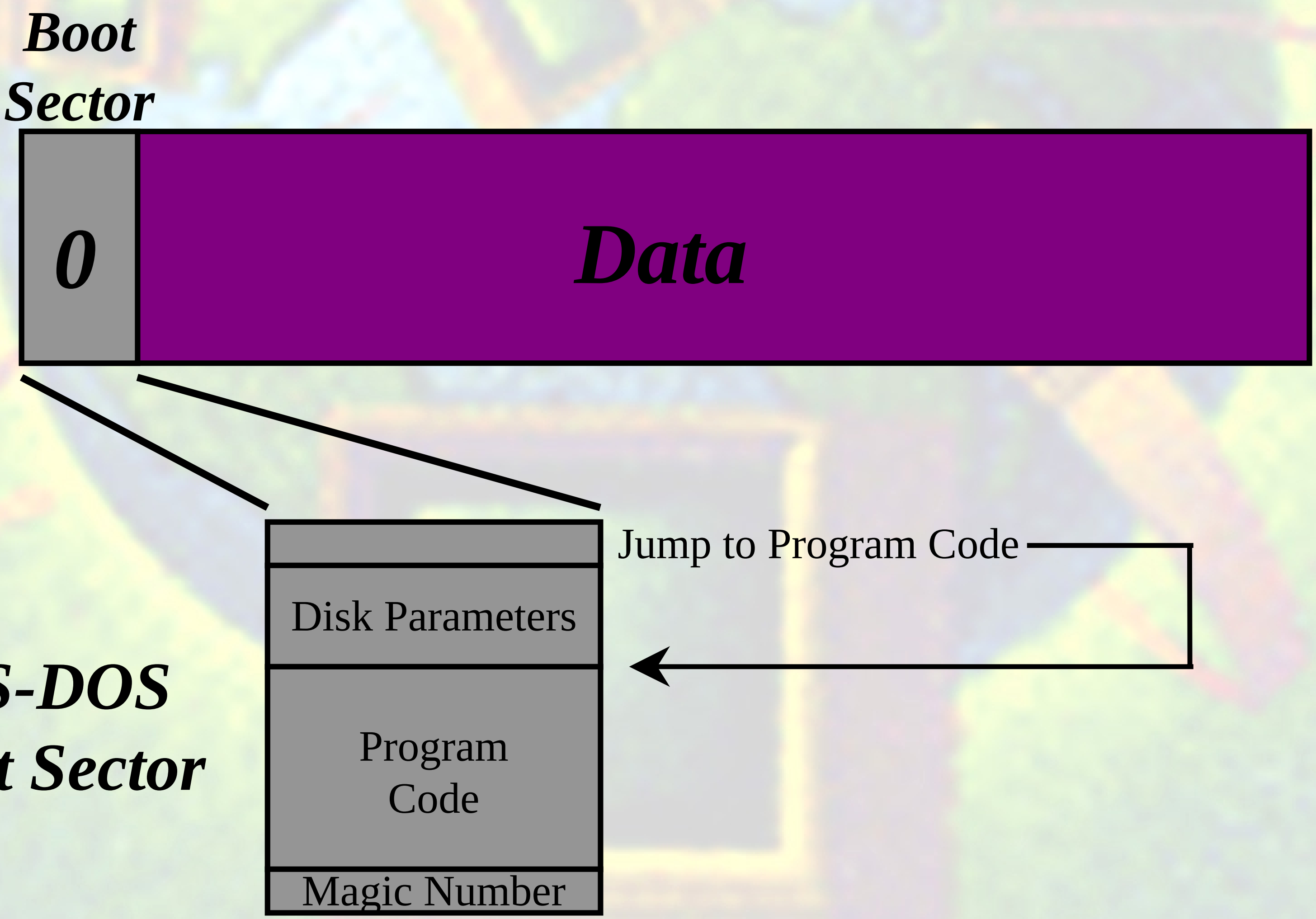
- LILO is a fairly complex topic, it can boot other operating systems, like Windows
- We will only introduce LILO, but there is a wealth of information in the LILO User's Guide
- Because of BIOS limitations, LILO's boot program may only be located in certain places:
  - The boot sector of a floppy
  - The master boot record (MBR) of the first hard disk *(the first IDE drive or SCSI address 0)*
  - The boot sector of a primary Linux file system partition on the first hard drive
  - A partition boot sector of an extended partition on the first hard drive

The Linux Loader: LILO

(continued)

- LILO is not the only way to boot a Linux system
  - Other Linux boot managers (bootactv, loadlin, etc.)
  - Commercially available boot managers (i.e. Partition Magic)
  - Windows NT boot manager
- The important thing to remember: whatever boot manager is in control must be able to find and access the boot sector containing LILO's loader
- If your system's hard drive configuration does not meet the Linux booting requirements, then you can boot from a floppy disk
- If your system is ONLY Linux, then things are much simpler!

Data Layout for a Floppy Disk



Boot Sector

0

Data

Jump to Program Code

Disk Parameters

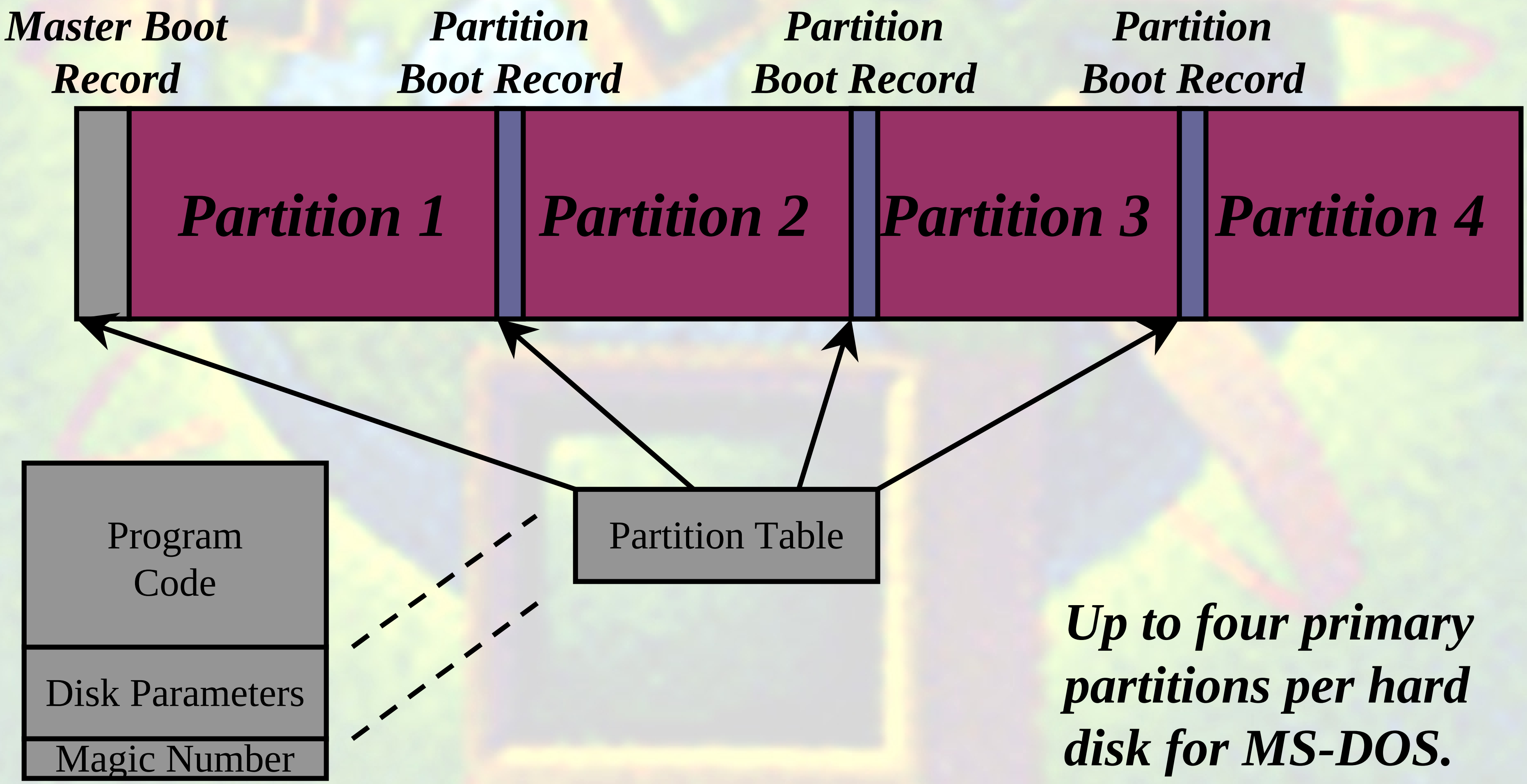
Program Code

Magic Number

MS-DOS Boot Sector

Data Layout for a Hard Disk

(without extended partitions)



Master Boot Record

Partition Boot Record

Partition Boot Record

Partition Boot Record

Partition 1 | ***Partition 2*** | ***Partition 3*** | ***Partition 4***

Partition Table

Program Code

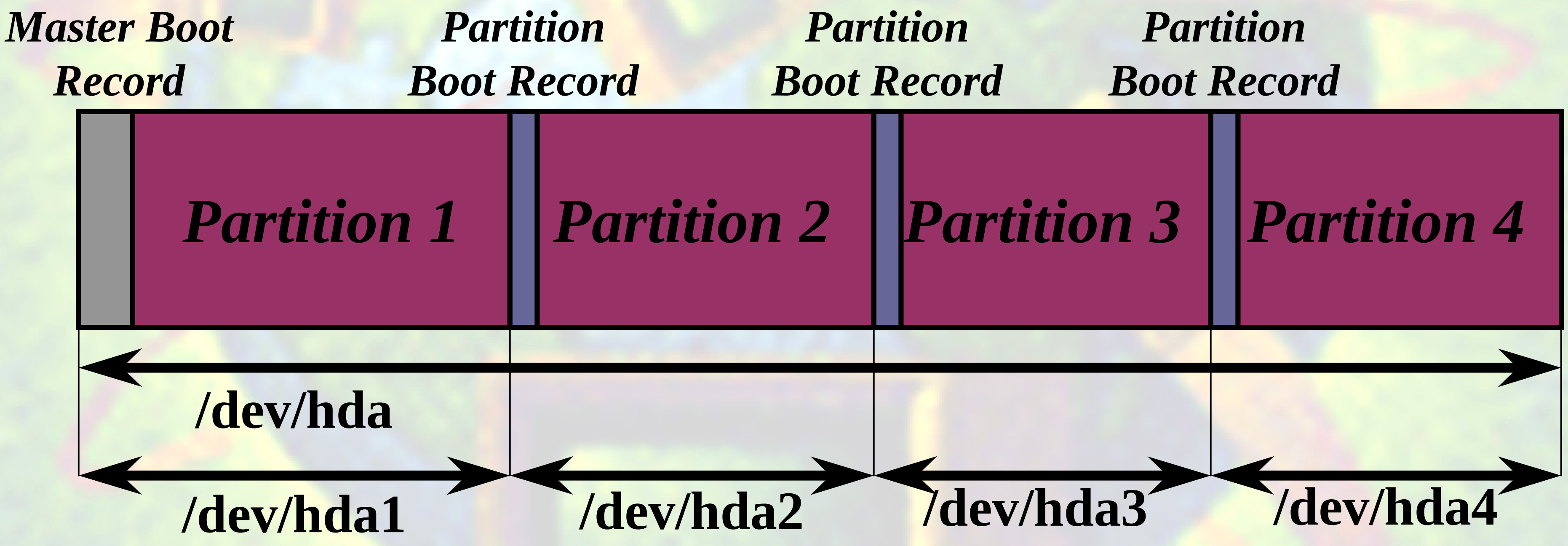
Disk Parameters

Magic Number

Up to four primary partitions per hard disk for MS-DOS.

Data Layout for a Hard Disk

(without extended partitions)



Master Boot Record

Partition Boot Record

Partition Boot Record

Partition Boot Record

Partition 1

Partition 2

Partition 3

Partition 4

/dev/hda

/dev/hda1

/dev/hda2

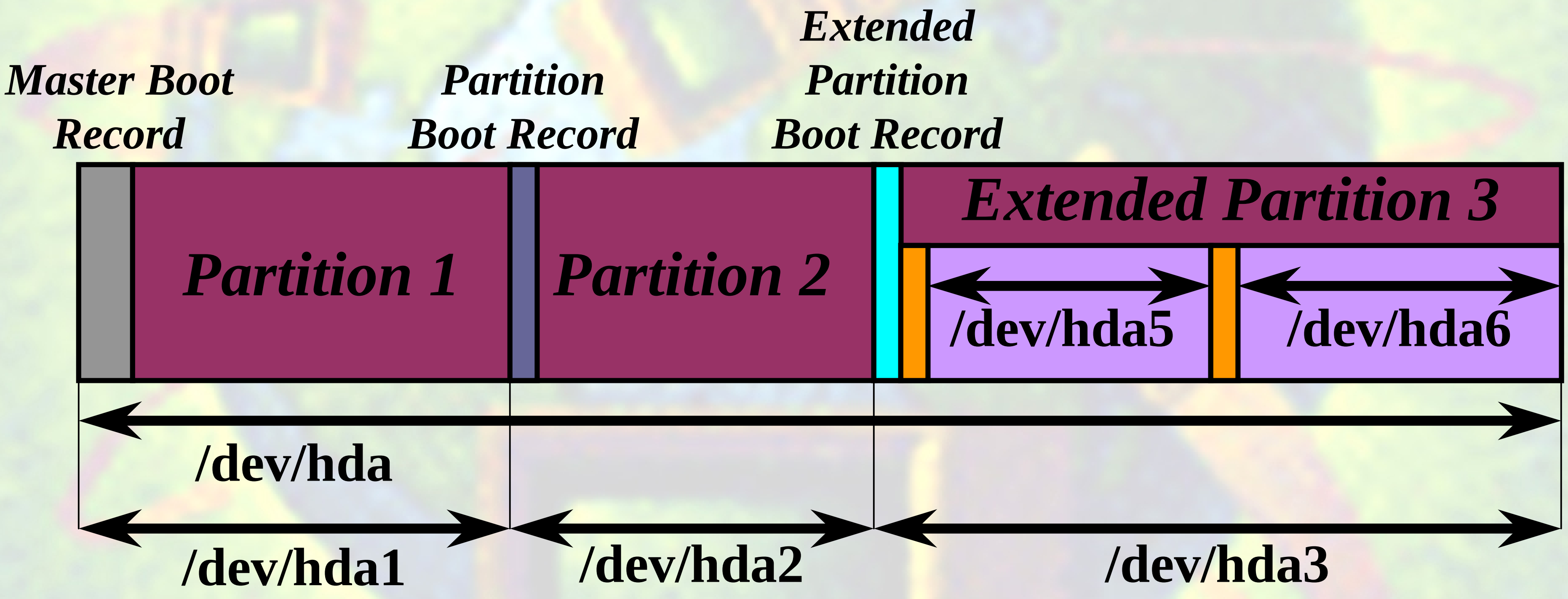
/dev/hda3

/dev/hda4

Note that each partition is accessible as if it were an independent device with its own boot sector. Linux creates devices for this configuration as shown.

Data Layout for a Hard Disk

(with extended partitions)



Master Boot Record

Partition Boot Record

Extended Partition Boot Record

Partition 1

Partition 2

Extended Partition 3

/dev/hda5

/dev/hda6

/dev/hda

/dev/hda1

/dev/hda2

/dev/hda3

If more than four partitions are needed, one primary partition is divided into an extended partition containing several logical partitions. Note that the partition tables of the logical partitions are not accessible as the first block of some device.

Lessons Learned About Installation and Booting

- Always make a boot disk!
- Try to avoid replacing the Windows MBR with LILO on a multiple boot machine (get a boot manager instead)
- Save the boot sectors that are replaced by LILO installation (/boot/boot.NNNN) in case you need to restore them
- Always make a boot disk!
- You can place the /boot partition into a small partition on the first IDE/SCSI drive and still mount the Linux root partition from elsewhere
- ALWAYS MAKE A BOOT DISK!

Contents of the /boot Directory

- If you look into a Linux /boot directory, you might see:

```
system.map -> system.map-2.2.12-20
system.map-2.2.12-20
initrd-2.2.12-20
map
module-info -> module-info-2.2.12-20
module-info-2.2.12-20
vmlinux-2.2.12-20
vmlinuz -> vmlinuz-2.2.12-20
vmlinuz-2.2.12-20
```

- On an SMP system, the files might have an “smp” suffix (vmlinux-2.2.12-20smp, etc.)

Passing Parameters to the Linux Kernel

- The Linux kernel, just like HP-UX, may be passed "command-line" parameters when it is started (remember the "hpux -is disk(;0)vmuni command?)
- LILO can pass the following parameters (and many more):
  - **read-only** **mount the root read-only**
  - **vga=mode** **normal(80x25), extended(80x50), ask**
  - **ramdisk=<size>**
  - **root=<root_device>** **device name or "current"**
  - **append=<string>** **append <string> to options**
  - **literal=<string>** **override ALL options with <string>**
- **LILO input: "<name> single" will boot into single-user mode**

More Lessons About LILO

- When you update a kernel, you must re-run LILO
- You should uninstall LILO before "decommissioning" a system
- The default configuration file for LILO is ***/etc/lilo.conf***
- You can build a "mini" boot partition on a floppy, either by hand or using the "/sbin/mkbootdisk" command
- The LILO command may be run on a mounted disk by specifying the "-r <directory>" option, which will do a "chroot" to that directory
- Examine the "mkbootdisk" script for examples of what is needed to build a bootable floppy disk

An Example LILO Configuration File

```
boot      = /dev/fd0                          # Specify boot device
delay     = 10                                # Wait 10 seconds
message = bootmessage                         # Text prompt
read-only                                     # Mount root RO

label = linux_up                              # Uniprocessor
          image = vmlinuz-2.2.12-20
          initrd = initrd-2.2.12-20.img
          root    = /dev/hda5
label = linux_smp                             # Multi-processor
          image=vmlinuz-2.2.12-20smp
          initrd = initrd-2.2.12-20smp.img
          root   = /dev/hda5
label = linux_old                             # Last week’s kernel
          image = vmlinuz_old
          initrd = initrd-2.2.12-20old.img
          root   = /dev/hda5
```

Sample System Hardware Configuration

Description	*Use*	*Device Name*
IDE Drive 1	Win98/Linux	/dev/hda
Windows partition	Windows 98	/dev/hda1
Linux partition	/boot	/dev/hda2
NEC CD-ROM stacker		/dev/hdc
HP R/W CD-ROM		/dev/hdd
Floppy drive		/dev/fd0
SCSI address 0	swap	/dev/sda1
SCSI address 0	Linux /	/dev/sda5
SCSI address 1	Linux /aux	/dev/sdb1
SCSI address 1	swap	/dev/sdb5

Sample System Hardware Configuration *(continued)*

Description	*Use*	*Device Name*
SCSI address 2	Linux /vmdata1	/dev/sdc1
SCSI address 2	swap	/dev/sdc5
SCSI address 3	Linux /vmdata2	/dev/sdd1
SCSI address 3	swap	/dev/sdd5
SCSI address 4	HP Photo scanner	/dev/sge
SCSI address 5	HP 6200C scanner	/dev/sgf
SCSI address 6	JAZ 1 GB drive	/dev/sde4
COM1	V.90 modem	/dev/ttyS0
Parallel port	HP P1000 printer	/dev/lp0

Sample /etc/fstab for Example Hardware

```
/dev/sda5     /              ext2     defaults             1  1
/dev/hda2     /boot          ext2     defaults             1  2
/dev/sdb1     /aux1          ext2     defaults             1  2
/dev/sdc1     /vmdata1       ext2     defaults             1  2
/dev/sdd1     /vmdata2       ext2     defaults             1  2
/dev/sda1     swap           swap     pri=1                0  0
/dev/sdb5     swap           swap     pri=1                0  0
/dev/sdc5     swap           swap     pri=1                0  0
/dev/sdd5     swap           swap     pri=1                0  0
/dev/fd0      /mnt/floppy    ext2     owner,noauto         0  0
/dev/cdrom    /mnt/cdrom     iso9660  owner,noauto,ro      0  0
/dev/hdd      /mnt/cd-rw     iso9660  noauto,ro            0  0
/dev/sde4     /mnt/jaz       vfat     fat=16               0  0
/dev/hda1     /mnt/Win98     vfat     fat=32               0  0
none          /proc          proc     defaults             0  0
none          /dev/pts       devpts   gid=5,mode=620       0  0
```

Linux Kernel Modules

- When configuring a kernel, you have the choice to build things in (monolithic) or to dynamically load modules (slower) but much easier …
- The "lsmod" command lists all loaded modules and their current state
- The startup operation will detect hardware and load the proper module (usually) for any *supported* hardware
- At boot time, the startup process runs "depmod -a", which builds a dependency list for all modules
- The /etc/conf.modules file contains commands for the module commands "insmod" and "modprobe"
- Unused modules are unloaded by a cron job which executes "rmmod -a"
- Modules are located under "/lib/modules/<kernel_rev>"

Building the Linux Kernel

- Note: Redhat Linux is compiled for maximum compatibility, for an i386 processor!
- To build the kernel:
  - cd /usr/src/linux (a link to the current version)
  - Build the configurator:
    - make xconfig (X-windows configurator tool)
    - make menuconfig (VGA mode configurator tool)
  - Save the current configuration to a file!
  - Make changes using the conrigurator
  - Save new configuration to a file!
  - Build the kernel (Be prepared for a wait…)
    - make deps (build dependencies)
    - make bzImage (a compressed kernel)
    - make modules (if you are using them)

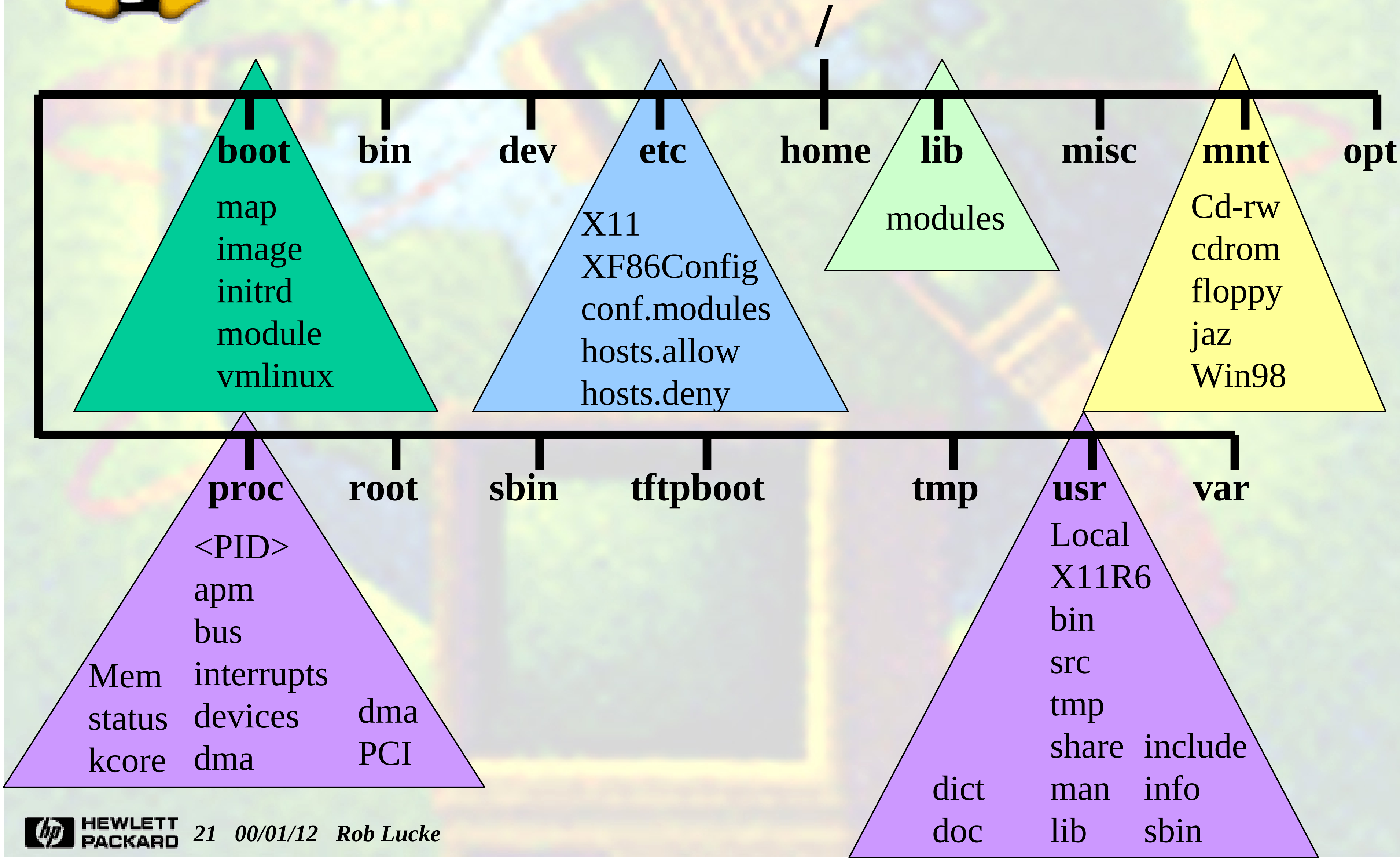
Building the Linux Kernel

(continued)

- Back up the current module information!
- Backup the current kernel and system files!
- Always make a boot disk! (/sbin/mkbootdisk)
- make modules_install (install new modules)
- Move kernel and other system files from /usr/src/linux/arch/i386/boot, this will be called "bzImage"
  - Can use /sbin/installkernel <version> <boot> <map>
- Run LILO to update map information (offset into partition for kernel file) in boot record
- Reboot and test

- Naming conventions are important
  - Add a "test" label to the LILO configuration file?
  - With links and "generic" LILO labels, you can implement a relatively fail-safe way of installing new kernels

Basic Linux File-system Structure



/

boot

map
image
initrd
module
vmlinux

bin

dev

etc

X11
XF86Config
conf.modules
hosts.allow
hosts.deny

home

lib

modules

misc

mnt

Cd-rw
cdrom
floppy
jaz
Win98

opt

proc

<PID>
apm
bus
interrupts
devices
dma

Mem
status
kcore

dma
PCI

root

sbin

tftpboot

tmp

usr

Local
X11R6
bin
src
tmp
share
man
lib

include
info
sbin

dict
doc

var

Some Basic Linux Command Differences

HP-UX:
UNIX
for the Enterprise

swapinfo	swapon -s
chown root:root /root_home	chown root.root /dev/fd0
rm	rm -f (defaults to "safe")
ll	alias ll='ls -al'
bdf	df

- Note: Many Linux commands support both the "-v" option style and the GNU "--verbose" option style. Check the man page for the command for details

Init and Run-Levels

- Init run levels:
  - 0 halt
  - 1 Single-user mode
  - 2 Multi-user without NFS
  - 3 Full multi-user
  - 4 Unused (really?)
  - 5 X11
  - 6 Reboot
- The “id:5:initdefault:” line in /etc/inittab controls default level
- The “runlevel” command returns previous level and current level (N=none) “N 5”
- Run-level information is available in two shell environment variables: ${RUNLEVEL} and ${PREVLEVEL}
- “init <level>” will change the run level

Linux System Start-up

- Similar to HP-UX, but:
  - /etc/rc.d contains
    - init.d (scripts)
    - rc0.d through rc6.d (links to scripts)
    - sysconfig (data for scripts)
- Startup is begun by the “init” process
- /etc/rc.d/sysinit is the initial script (pun)
- /etc/rc
- /etc/rc.local

Interesting Side Trips

- /etc/pam.d directory containing PAM config
- /etc/profile.d directory containing shell profiles
- /etc/logrotate.d directory controlling log rotation
  - /usr/sbin/logrotate command
  - /etc/logrotate.conf
- /etc/rc.d directory containing startup scripts
- /proc directory containing extensive system/process information

Linux Shells (/etc/shells)

• /bin/bash	"GNU Born-again shell"
• /bin/sh	Link to /bin/bash
• /bin/ash	"sh with System V features"
• /bin/bsh	Link to /bin/ash
• /bin/tcsh	"C shell with enhancements"
• /bin/csh	"C shell"
• /bin/bashZ	Link to /bin/bash
• /bin/ksh	Korn shell
• /bin/zsh	"Like ksh, w/many enhancements"

Configuring PPP on Linux

- Determine your modem device (/dev/ttyS0)
- The GNOME control panel
  - Add Applet ->
  - Network ->
  - RH PPP Dialer (select)
- Right click on applet when it appears
  - configure PPP (and “debug” to test configuration)
    - modem tab (configure your modem)
    - accounts tab (add an account for your ISP)
- Modifies the /etc/wvdial.conf file used by /usr/bin/wvdial (see man wvdial)
- Connect using applet or “wvdial MyISP”

Example /etc/wvdialconf File

```
[Modem0]
Modem = /dev/ttyS0
Baud= 115200
Init1 = ATZ
SetVolume = 1
Dial Command = ATDT
Init4 = ATM1L1

[Dialer Defaults]
Modem = /dev/ttyS0
Baud = 115200
Init1 = ATZ
SetVolume = 1
Dial Command = ATDT
Init4 = ATM1L1

[Dialer MyISP_V90]
Username = Myname
Password = MyPassword
Modem = /dev/ttyS0
Init1 = AT &F E0 &C1 &D2 V1 S0=0\V1
Init2 = ATs7=60S30=60L0M1\N3%C1&K3B0B15N1X4
Phone = 555-1212
Baud = 115200
Stupid mode = 0
Inherits = Modem0

[Dialer ppp0]
Modem = /dev/modem
Baud = 115200
Stupid mode = 0
```

Linux Password Management

- Shadow password file is the default, no passwords are available to casual users
  - Too much password data kept to go into here…
  - rob:x:1000:100:/home/rob:/bin/bash
  - chage   Change user expiration information
  - pwck   Check password integrity
  - grpck   Check group integrity
- /etc/passwd /etc/passwd-   /etc/group /etc/group-
- /etc/shadow   (mode 400, root:root)
- /etc/gshadow   (mode 400, root:root)
- passwd -> pwconv -> shadow
- shadow -> pwunconv -> passwd
- group -> grconv -> gshadow
- gshadow -> grunconv -> group

Linux Security

- Linux, by default is setup with more security features enabled than HP-UX
- You will not be able to log in as “root” except at the console
- Telnet in as a “normal” user, then “su -” to “root”, but you must HAVE a user configured besides “root”…
- Check out /etc/hosts.allow and /etc/hosts.deny to configure machines that can access your Linux box

System Management Tools

- The major system management tool for Redhat Linux is called “linuxconf”
- Menu and check-box driven configuration of Linux features
- No changes are made until “committed” -- remember this, or you will be surprised!
- Still a little cryptic and opaque to me

- I never thought I would compliment the HP-UX SAM utility, but…

Linux Miscellaneous

- Two desktops:

– GNOME	GNU Network Object Model Environment
– KDE	K Desktop Environment

- Help in Linux:

– help	shell help
– man	vi man page reader
– info	emacs info (remember ^X^C to exit…)

- Editors

– vi-> vim	Improved vi
– emacs	GNU advanced editor
– others	Too many to count… .

Summary of Linux

- This has not been a comprehensive survey of all system features, there is a LOT there!
- Plan on re-installing the system a lot and making lots of mistakes (I did)
- I have shown you the tips of SOME of the icebergs, now put on your wetsuit!
- Enjoy Linux! It is what Unix used to be and what it will be again, all at once.