Introduction to Unix

AfNOG 2007 Workshop

April 22, 2007 - Abuja, Nigeria

Hervey Allen
Phil Regnauld

Network Startup Resource Center

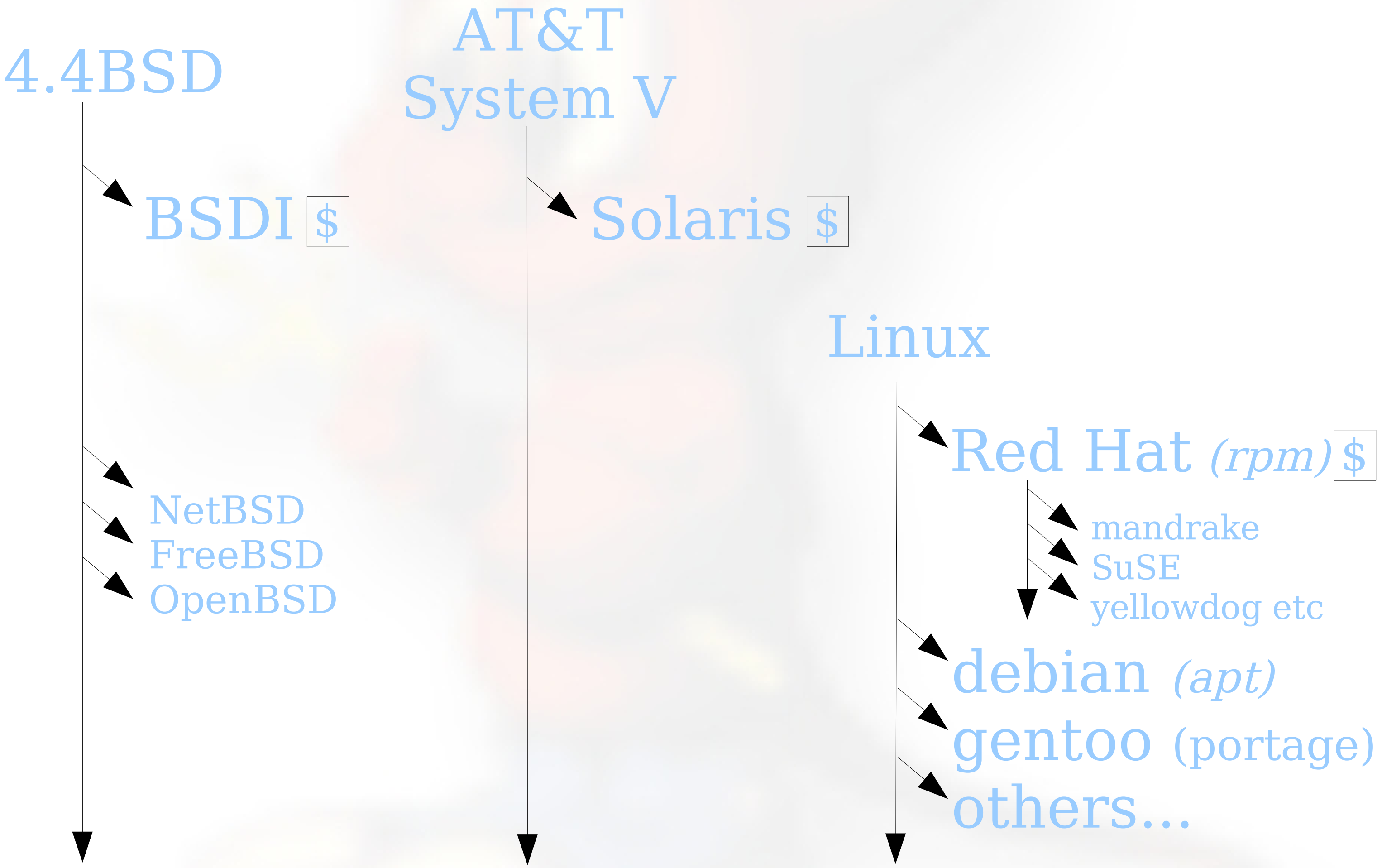
What's Our Goal?

- To introduce basic UNIX concepts
- To present FreeBSD, a UNIX flavor
- To practice some of the concepts with exercises.
- To give you the fundamental concepts and practice needed in UNIX to be able to concentrate on what's being taught during the week.

Some Practical Matters

- *Please do not change the root password.*
- Please do ask questions! Lots of questions! Really -we mean this.
- If you don't understand something be sure you ask for help! This is how you learn.
- Questions?

Simplified Unix family tree (Look at the wall... :-))



4.4BSD

BSDI $

NetBSD
FreeBSD
OpenBSD

AT&T
System V

Solaris $

Linux

Red Hat *(rpm)* $

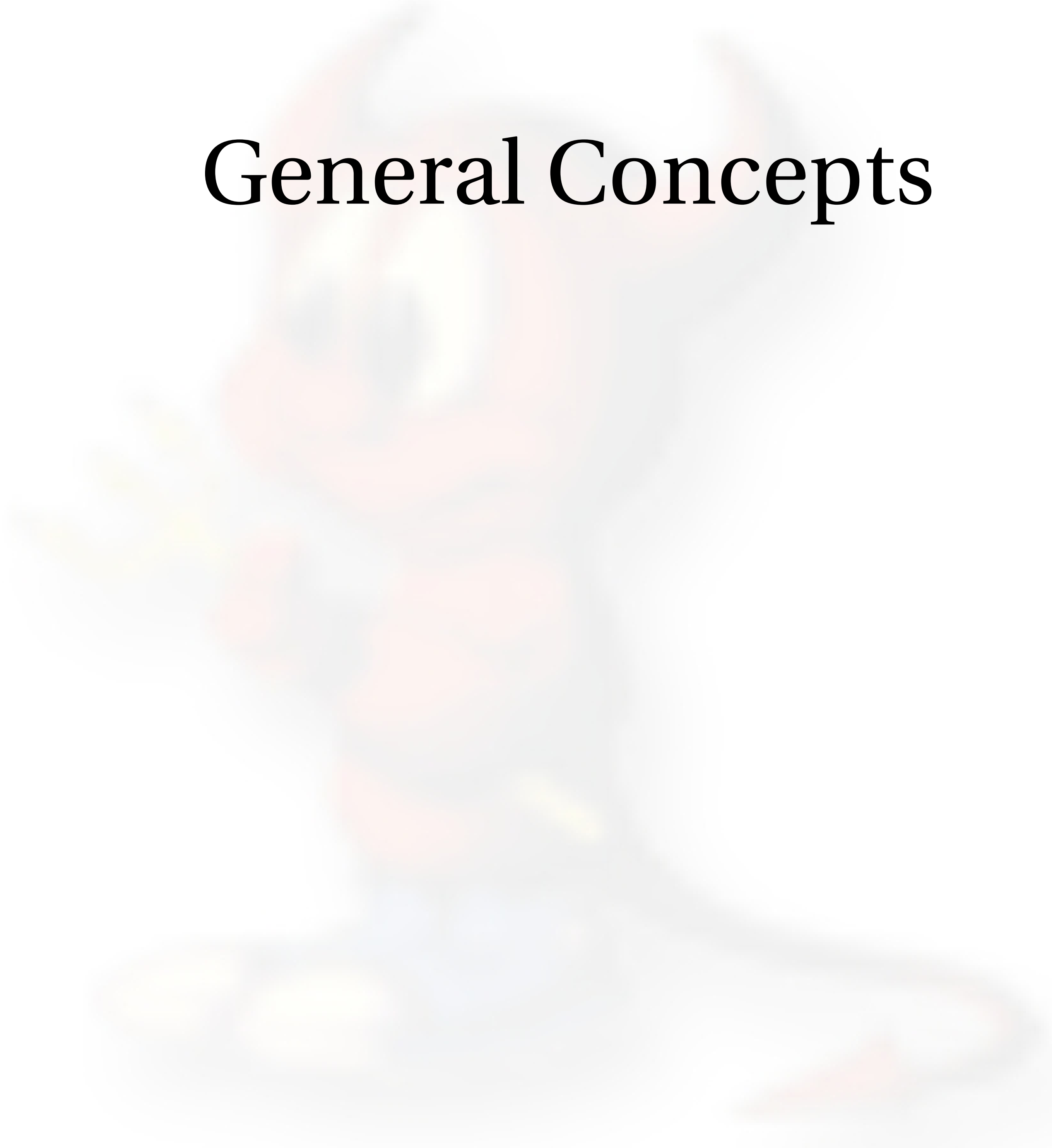
mandrake
SuSE
yellowdog etc

debian *(apt)*

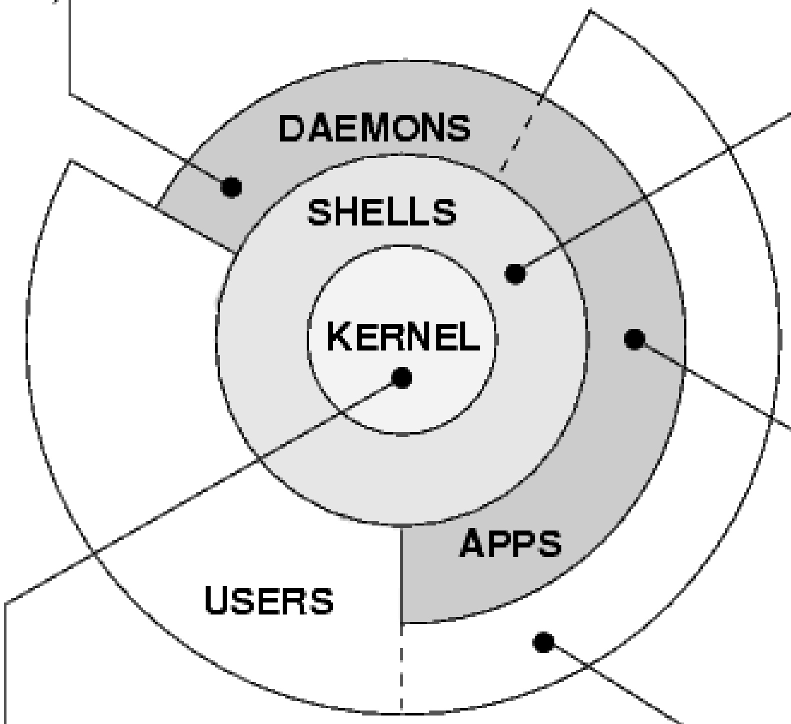
gentoo (portage)

others...

General Concepts



The UNIX system



Background processing
Non–interactive
(no terminal)

Launch processes
User interaction (terminal)
Job control
(scripting)

I/O (+ terminals)
Memory management (VM)
Interruptions
Scheduling / Timesharing
Networking

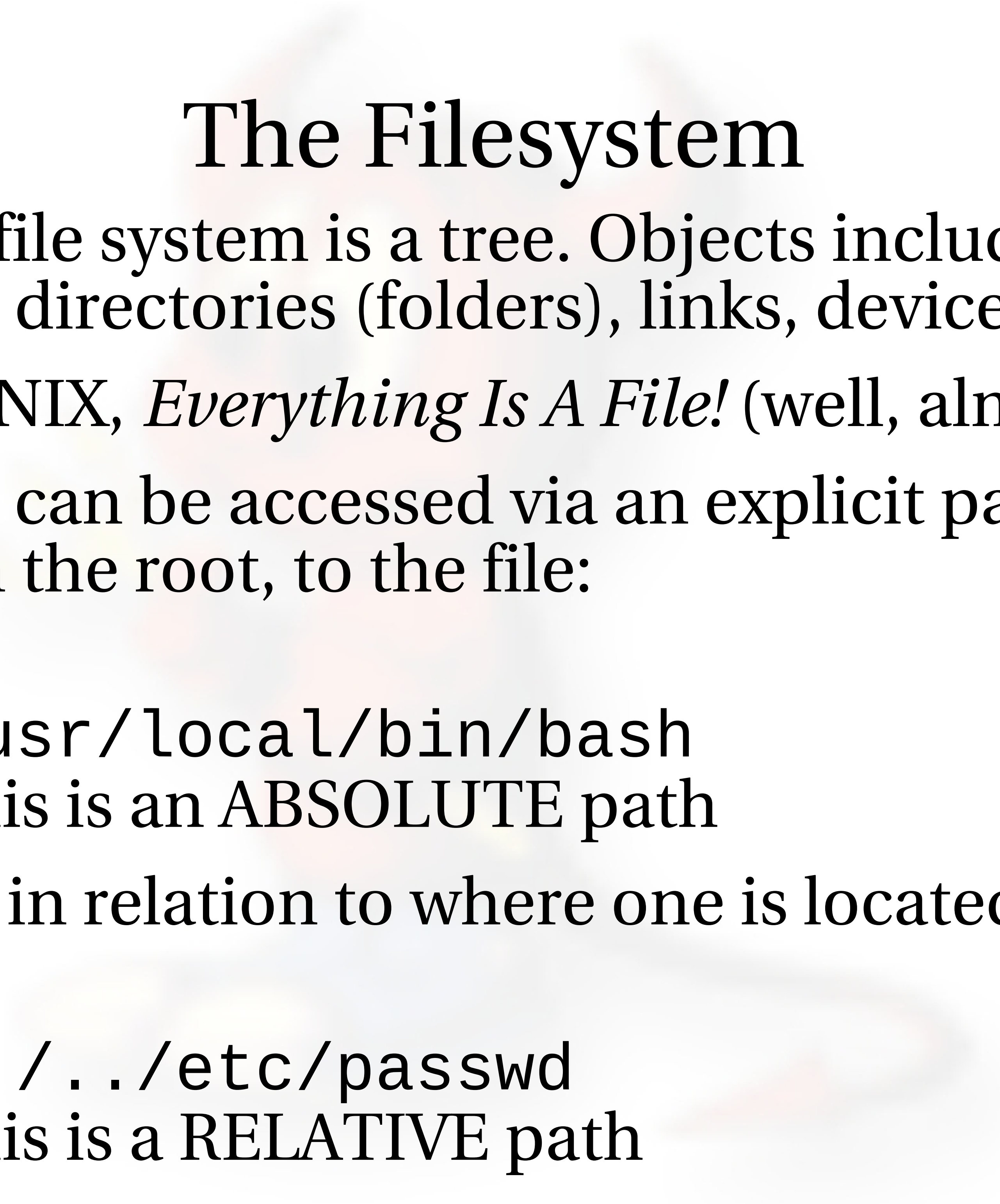
File management
Editors
Compilers
Network tools
...

Eat Resources :–)
(uid / gid)

The Kernel

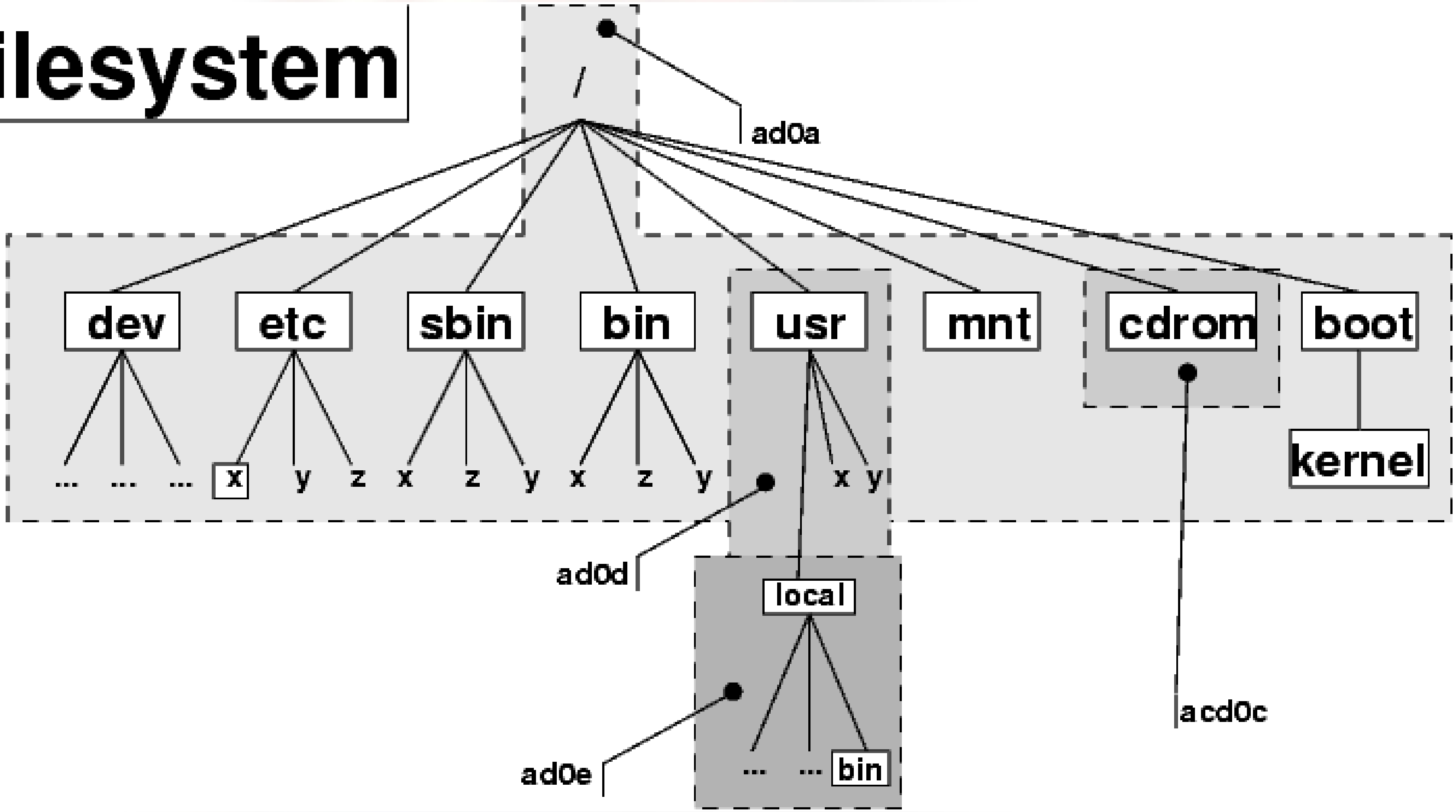
- Either monolithic or microkernel
- Takes care of managing resources and running processes in a fair way as well as enforcing rights and privileges
- Multiple programs (processes) can run concurrently thanks to the kernel
- Resource management includes: peripherals, memory, disk, network, CPU
- RAM + disk = virtual memory – kernel oversees memory access

The Filesystem



- The file system is a tree. Objects include files, directories (folders), links, devices
- In UNIX, *Everything Is A File!* (well, almost)
- Files can be accessed via an explicit path, from the root, to the file:
-
  ```
  /usr/local/bin/bash
  ```
  -> this is an ABSOLUTE path
- ... or in relation to where one is located:
-
  ```
  ../../etc/passwd
  ```
  -> this is a RELATIVE path

UNIX filesystem



Access rights

- Files are owned by a *user* and a *group* (ownership)
- Files have permissions for the user, the group, and the others
- The permissions are *Read, Write* and *Execute* (R, W, X)
- The same applies to all files

Users and Groups

- UNIX understands Users and Groups
- A user can belong to several groups
- A file can belong to only one user and one group at a time
- A particular user, the superuser *“root”* has extra privileges
- Only root can change the ownership of a file

Basic Commands

- cp, cd*, ls, mkdir, mv, rm, cat, less, more, ps, kill, w and ***man***
  - (*built in command shell commands).
- Where are commands located?
- /bin, /usr/bin, /usr/local/bin, /sbin, /usr/sbin
  - The difference between “sbin”, “bin” and “ /usr”
- Glossary:
  - cp = copy
  - cd/chdir = cd/chdir
  - ls = dir
  - mkdir = mkdir
  - mv = move (before it was copy and delete/erase)
  - rm = del[ete] and/or erase

Exercises

1 Create user accounts
2 Practice with basic commands
3 Getting help (manpages, docs and FreeBSD)
4 General job control
5 Processes and stopping them
6 Viewing file contents
7 File and directory permissions

FreeBSD Unix

The World of FreeBSD

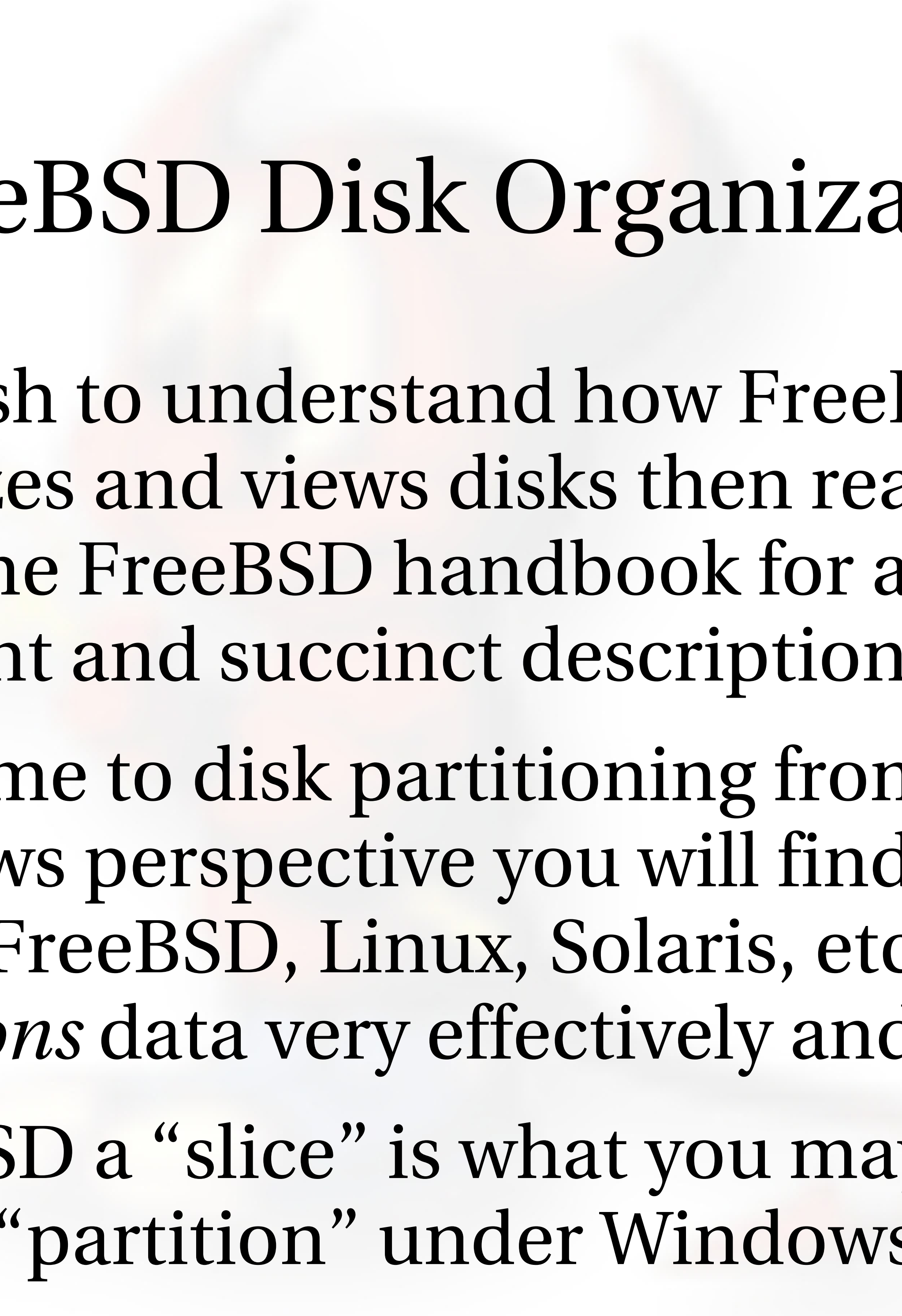
Start here: http://www.freebsd.org/

- RELEASE (5.5 legacy and 6.2)
- STABLE ('beta' code – like 6.2)
- CURRENT ('alpha' code)
- Documentation Project
  - FreeBSD Handbook

Installing FreeBSD

- How can you install? (FreeBSD Handbook section 2.2.6)
  - A CDROM or DVD
  - Floppy disks (including preconfigued install)
  - An FTP site, going through a firewall, or using an HTTP proxy, as necessary (with PXE)
  - An NFS server
  - A DOS partition on the same computer
  - A SCSI or QIC tape
  - A dedicated parallel or serial connection

FreeBSD Disk Organization



If you wish to understand how FreeBSD organizes and views disks then read section 3.5 of the FreeBSD handbook for an excellent and succinct description.

If you come to disk partitioning from a Windows perspective you will find that UNIX (FreeBSD, Linux, Solaris, etc.) *partitions* data very effectively and easily.

In FreeBSD a “slice” is what you may consider to be a “partition” under Windows.

FreeBSD Partition Schemes

Partition	Usage
a	Root partition (/)
b	swap partition
c	Not used for filesystems.
d	Supposedly not often used.
e/f	/tmp, /usr, etc…

View partition information using "`df  -h`" and "`swapinfo`"

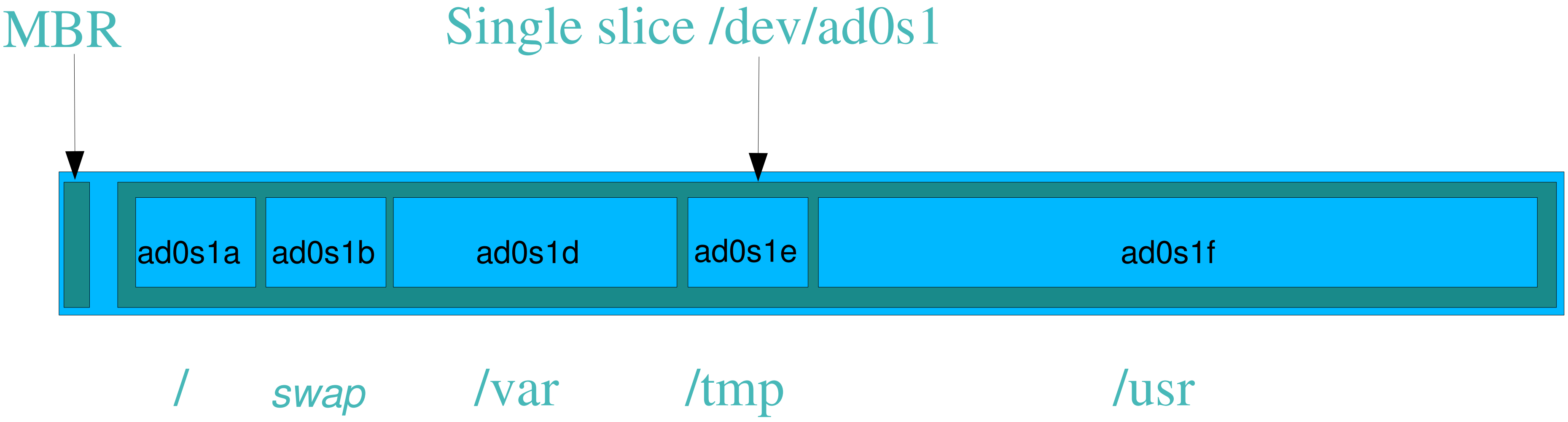
FreeBSD Disk Slices

Sample Output to view disk slices from “`fdisk -s`”

```
/dev/ad0: 77520 cyl 16 hd 63 sec
Part        Start        Size Type Flags
   1:          63     8385867 0x0b 0x80
   2:     8385930     8385930 0xa5 0x00
   3:    16771860      208845 0x83 0x00
   4:    16980705    61159455 0x0f 0x00
```

This is a 40GB disk with 3 operating systems spread across four slices. The operating systems include Windows 2000 (1), FreeBSD (2), Linux (3) and the 4th partition is a DOS swap slice for Windows 2000.

Simple partitioning: /dev/ad0



MBR

Single slice /dev/ad0s1

ad0s1a ad0s1b ad0s1d ad0s1e ad0s1f

/ *swap* /var /tmp /usr

/ (root partition)	ad0s1a	256MB
swap partition	ad0s1b	~ 2 x RAM
/var	ad0s1d	256MB (+)
/tmp	ad0s1e	256MB
/usr	ad0s1f	rest of disk

FreeBSD Partitions in a Slice cont.

To view slice partition information in a more "human" readable format use "`df -h`". This can, however, be misleading. For example:

```
Filesystem      Size    Used   Avail Capacity  Mounted on
/dev/ad1s1a     248M     35M    193M    15%    /
devfs           1.0K    1.0K      0B   100%    /dev
/dev/ad1s1e     248M    526K    227M     0%    /tmp
/dev/ad1s1f      55G    2.7G     48G     5%    /usr
/dev/ad1s1d     248M     42M    186M    18%    /var
/dev/ad1s2       55G     15G     38G    28%    /data
/dev/da0s1      500M    226M    274M    45%    /mnt/flash
```

Use "`swapinfo`" to see the swap partition:

```
Device          1K-blocks     Used    Avail Capacity
/dev/ad1s1b       1022784      124  1022660     0%
```

Command Line vs. GUI

- To administer a FreeBSD server you can do this entirely from the command line, or “shell”.
- A Graphical User Interface (GUI) is not necessary to provide services (web, email, print, file, database, etc.) using FreeBSD (or Linux/Unix).
- You can run multiple command line windows (shells) at the same time.
- To use a GUI you must install the X Windows system and a desktop environment such as Gnome or KDE.

Configuration via Files

- In the Windows world most configuration takes place inside the Windows Registry files. These are binary database files.
- Under FreeBSD (and Linux/Unix) almost all configuration is done using text files.
- Graphical tools to configure services under FreeBSD simply write to a configuration file.
- To configure services you usually need to be the system admin account, “root”, and you will often edit text files directly.

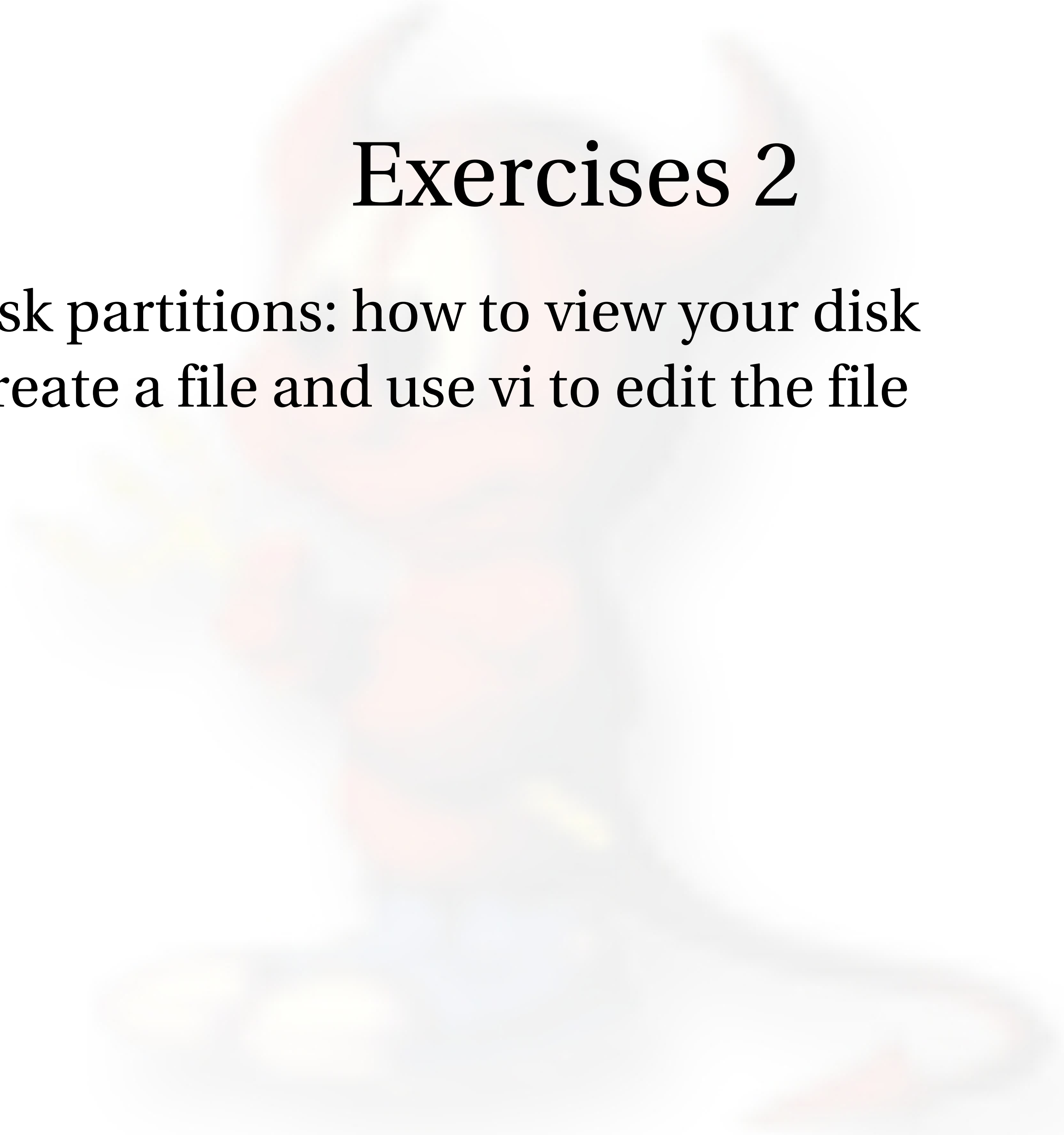
The vi Editor

- Why use vi? vi exists in almost all flavors of Unix and Linux.
- If you have to work on a new machine, then vi will almost always be available to you.
- In reality, you are likely to use a different editor for more complex editing, but we will practice using vi extensively today.

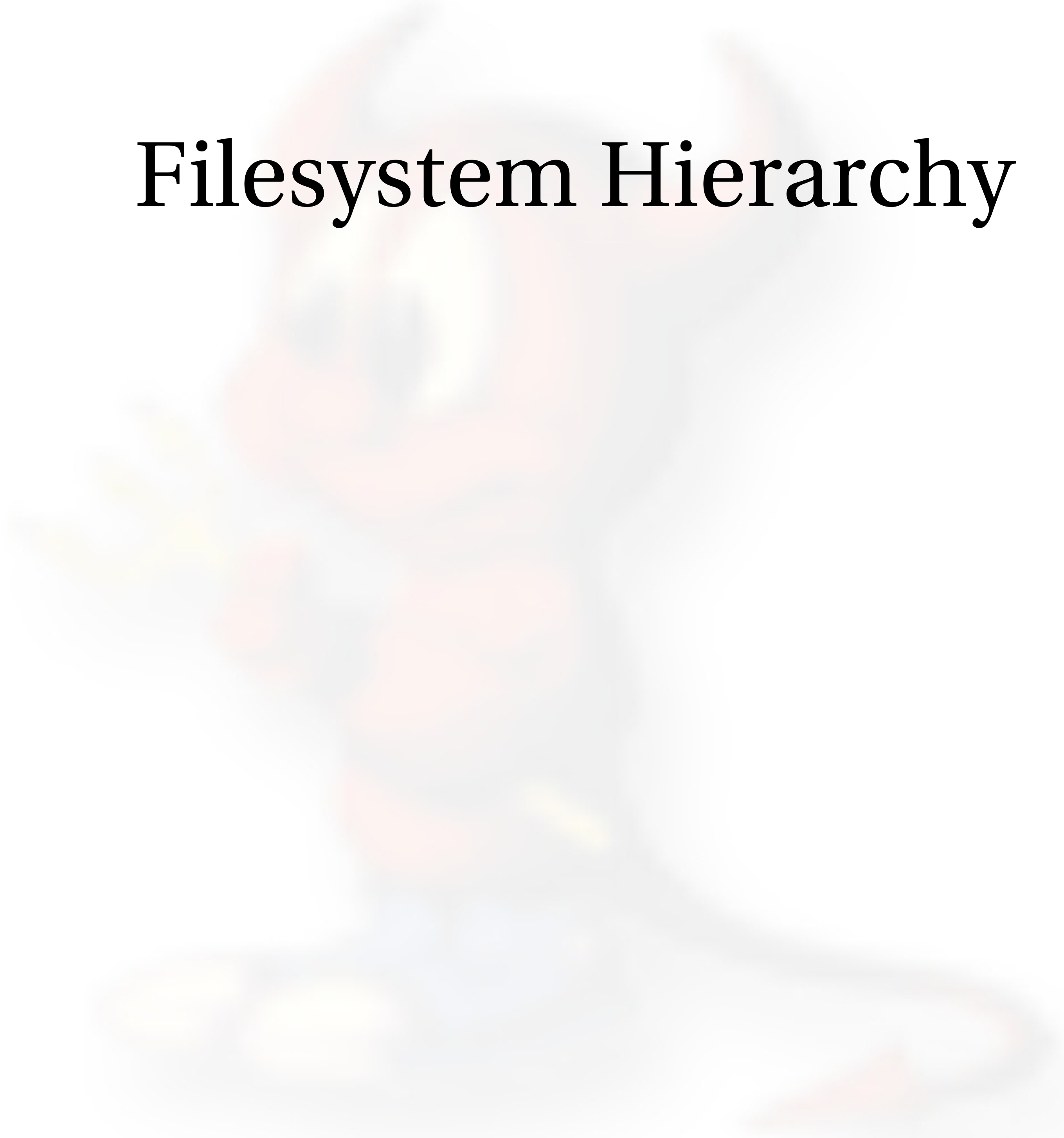
Exercises 2

8 Disk partitions: how to view your disk

9 Create a file and use vi to edit the file



Filesystem Hierarchy



FreeBSD Directory Structure

Repeat after me:
“The command `'man hier'` is your friend.”

So, why is your FreeBSD disk partition split in to “slices”? Largely to separate important file systems from each other. These filesystems are usually represented by specific directories.

Why not just run with everything in one place? That is, everything under root (`/`).

- Note: FreeBSD can optimize layout of files based on the use for the filesystem.

A Few FreeBSD Directories

- Structure of partitions/directories:
  - /   (“root”)
  - /usr
  - /var
  - swap

- Two important directories:
  - /var/tmp
  - /usr/home

“/” Root

The root partition is where critical system files live, including the programs necessary to boot the system in to “single user” mode.

The idea is that this part of the system does not grow or change, but rather stays isolated from the rest of the operating system.

If you give enough room to /usr and /var, then “/” can be quite small (around 512MB should be safe for now).

The one directory that may grow is /tmp, particularly if you run Linux binaries that use /tmp.

/usr

Is used for system software like user tools, compilers, XWindows, and local repositories under the /usr/local hierarchy.

If one has to expand* this partition for additional software, then having it separate makes this possible.

FreeBSD maps user directories to /usr/home.

*We'll discuss this. We don't always install FreeBSD with a separate /usr partition.

/var

This is where files and directories that consistently change are kept. For example, webserver logs, email directories, print spools, temporary files, etc.

On a server it is a good idea to have /var in a separate partition to avoid having it fill your other filesystems by accident.

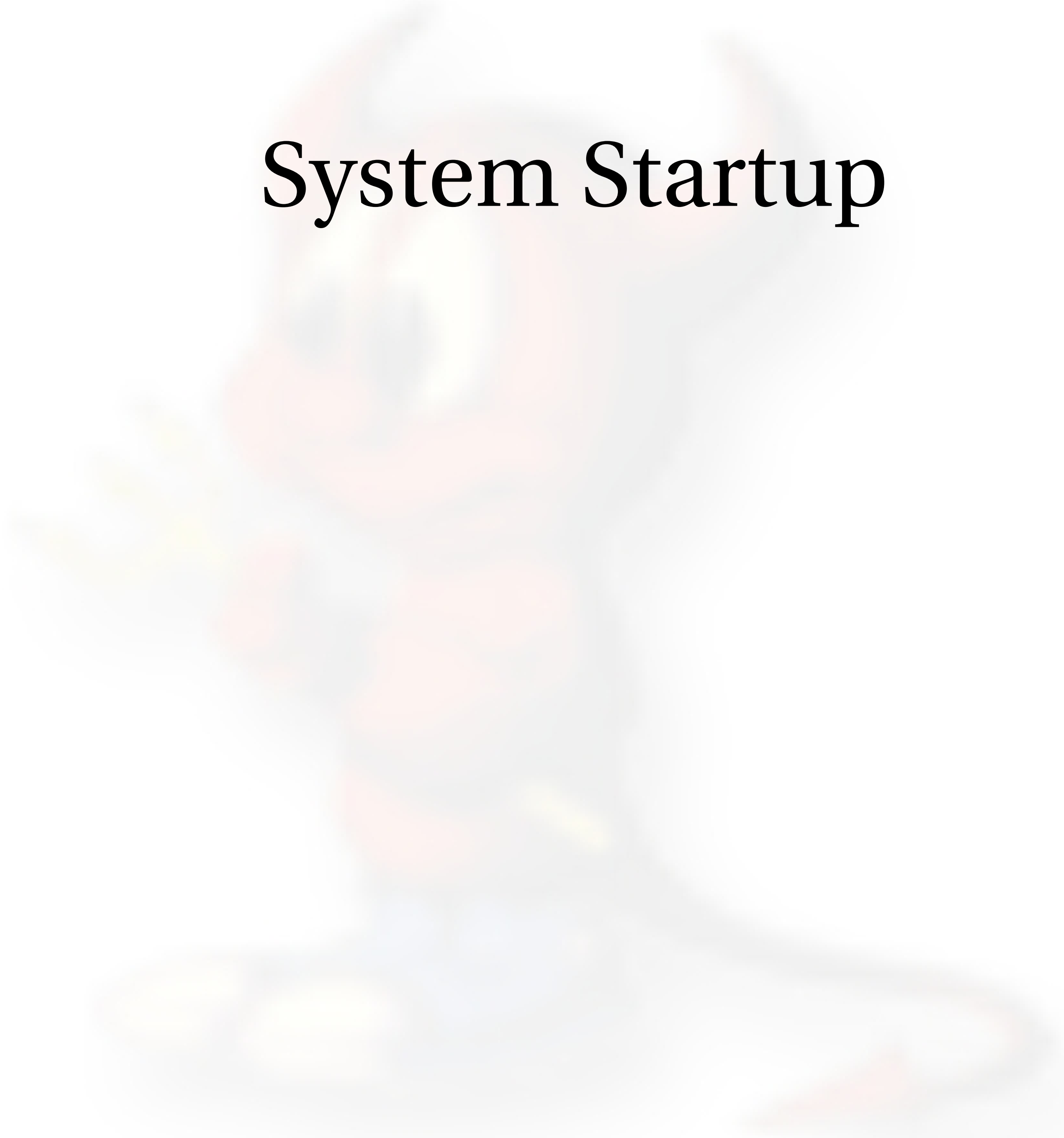
swap

Swap is where virtual memory lives. Swap is it's own filesystem.

You can run without swap, and your PC may run faster, but this is dangerous if you run out of memory.

There are several opinions about what is the optimal swap size. This can depend on what type of services you run (databases need more swap). The general rule of thumb is that swap size should be somewhere between your RAM and twice your server's RAM.

System Startup



How FreeBSD Boots

The init process:

- After the kernel boots, which is located in “/” (in Linux it's usually /boot) it hands over control to the program /sbin/init.
- If filesystems look good then init begins reading the resource configuration of the system. These files are read in this order:
  - /etc/defaults/rc.conf
  - /etc/rc.conf (overrides previous)
  - /etc/rc.conf.local (overrides previous)
- Mounts file systems in /etc/fstab

How FreeBSD Boots cont.

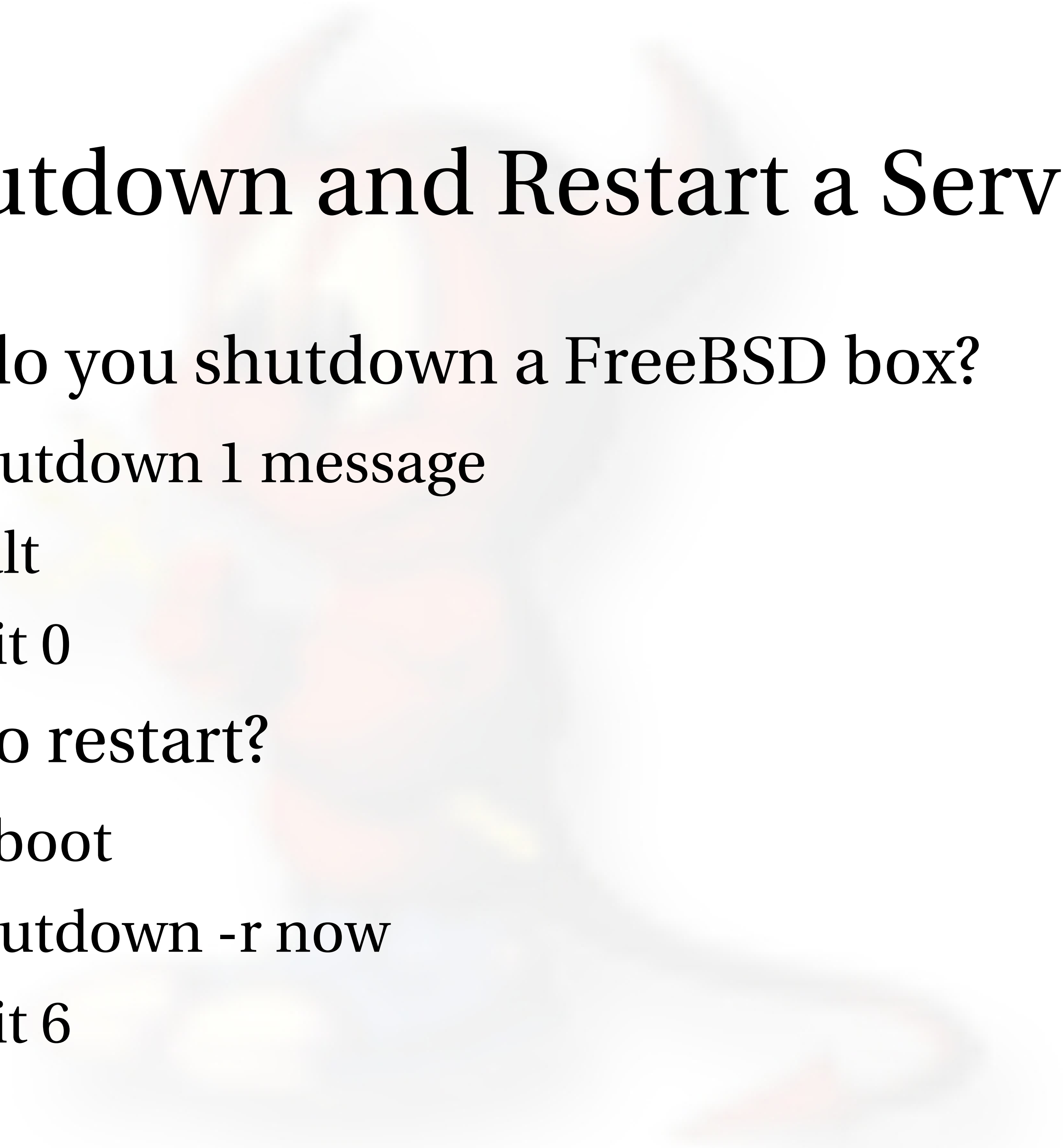
The init process cont.:

- Once file systems are mounted then the following starts:
  - Networking services
  - System daemons
  - Locally installed package daemons (/usr/local/etc/rc.d scripts)

Init process and shutdown:

- When `shutdown` is called then init runs the scripts /etc/rc.shutdown.

Shutdown and Restart a Server



How do you shutdown a FreeBSD box?

- shutdown 1 message
- halt
- init 0

And, to restart?

- reboot
- shutdown -r now
- init 6

Single vs. Multiuser Mode

FreeBSD has the concept of run levels:

Run-level	Signal	Action
0	SIGUSR2	Halt and turn the power off
1	SIGTERM	Go to single-user mode
6	SIGINT	Reboot the machine

So, in reality, you either run in single-user mode with “everything off” and just root access (“run-level 1”), or your system is up and fully running in multi-user mode.

To go from single-user to multi-user mode type “`exit`” at the command line.

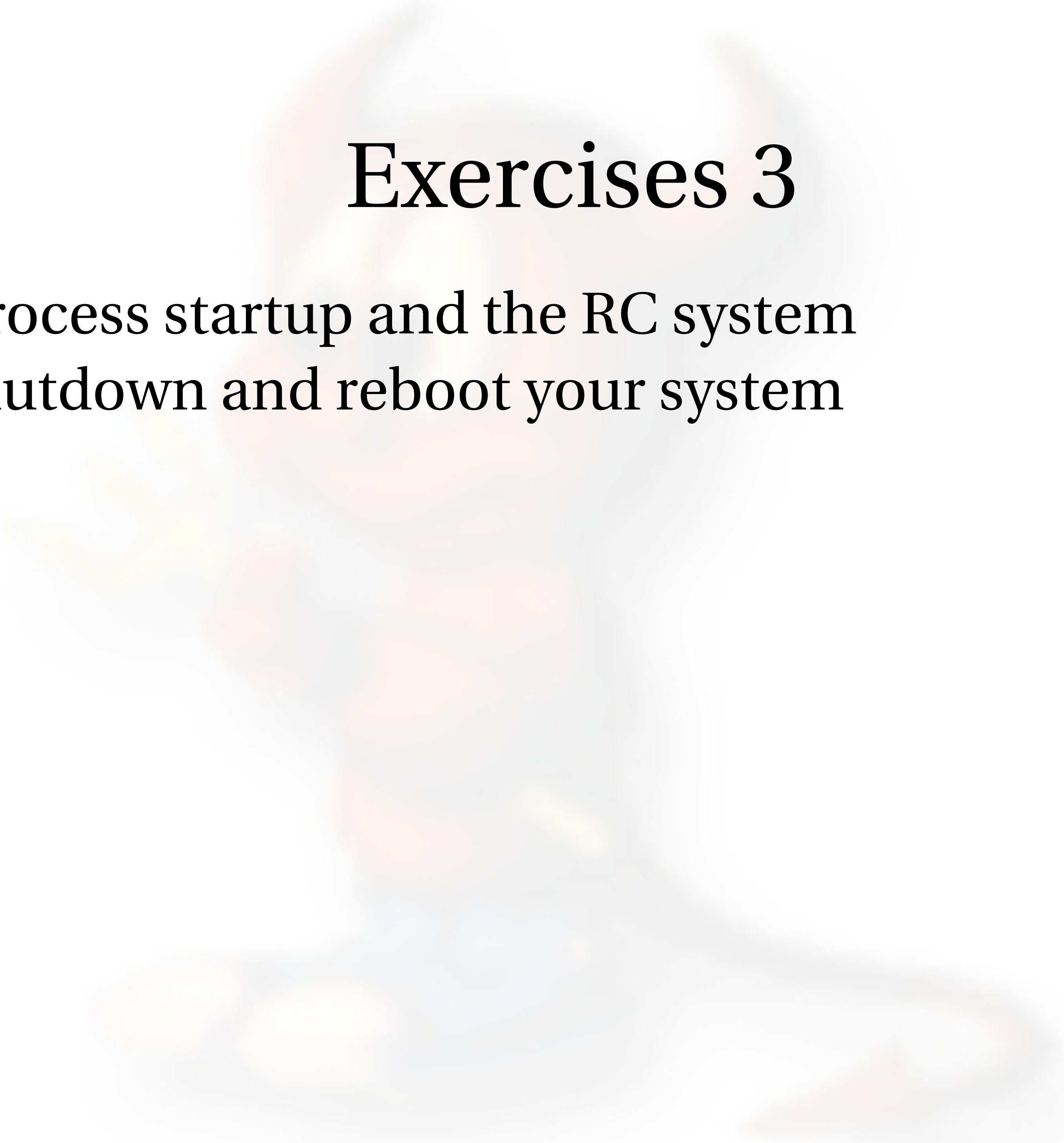
Starting/Stopping Services: Review

How does a service start/stop?

- `kill, /etc/rc.d/service stop`
- /etc/rc.d/service start ==> system
- /usr/local/etc/rc.d/script.sh ==> 3rd party
- /etc/rc.conf ==> system & some 3rd party
- /etc/defaults/rc.conf ==> leave alone
- Old school: /etc/rc.local
- Read “`man rc`” *several times*! :-)

Exercises 3

10 Process startup and the RC system
11Shutdown and reboot your system



Installing Software

Software Install Methods

There are three methods to install software on your FreeBSD system. These are:

1.) FreeBSD packages and the `pkg` utility.

2.) The ports collection /usr/ports.

3.) Installing from source (gcc make).

Using *portupgrade*, a meta software install utility, is recommended.

There are advantages and disadvantages to each. These are quite distinct from Windows.

The “pkg” Commands

In general the pkg_add and pkg_delete facilities allow you to install and remove software on your system in an efficient and consistent manner.

The pkg_info command allows you to see what's installed, quickly, and to get detailed information about each software package that is installed.

Installing from Ports

First you must have installed the /usr/ports collection during system installation. Otherwise, use /stand/sysinstall after installation and then choose Configure, Distributions, then Ports.

Once the “ports collection” is installed you can see the entire tree under /usr/ports. There are several thousand software packages available.

This collection contains minimal information so that you can “make” a software package quickly, and easily from separate CD-ROMs or a network site containing the port source.

See section section 4.5 of the FreeBSD Handbook.

Portupgrade

Portupgrade combines the “best of both world” from packages and ports.

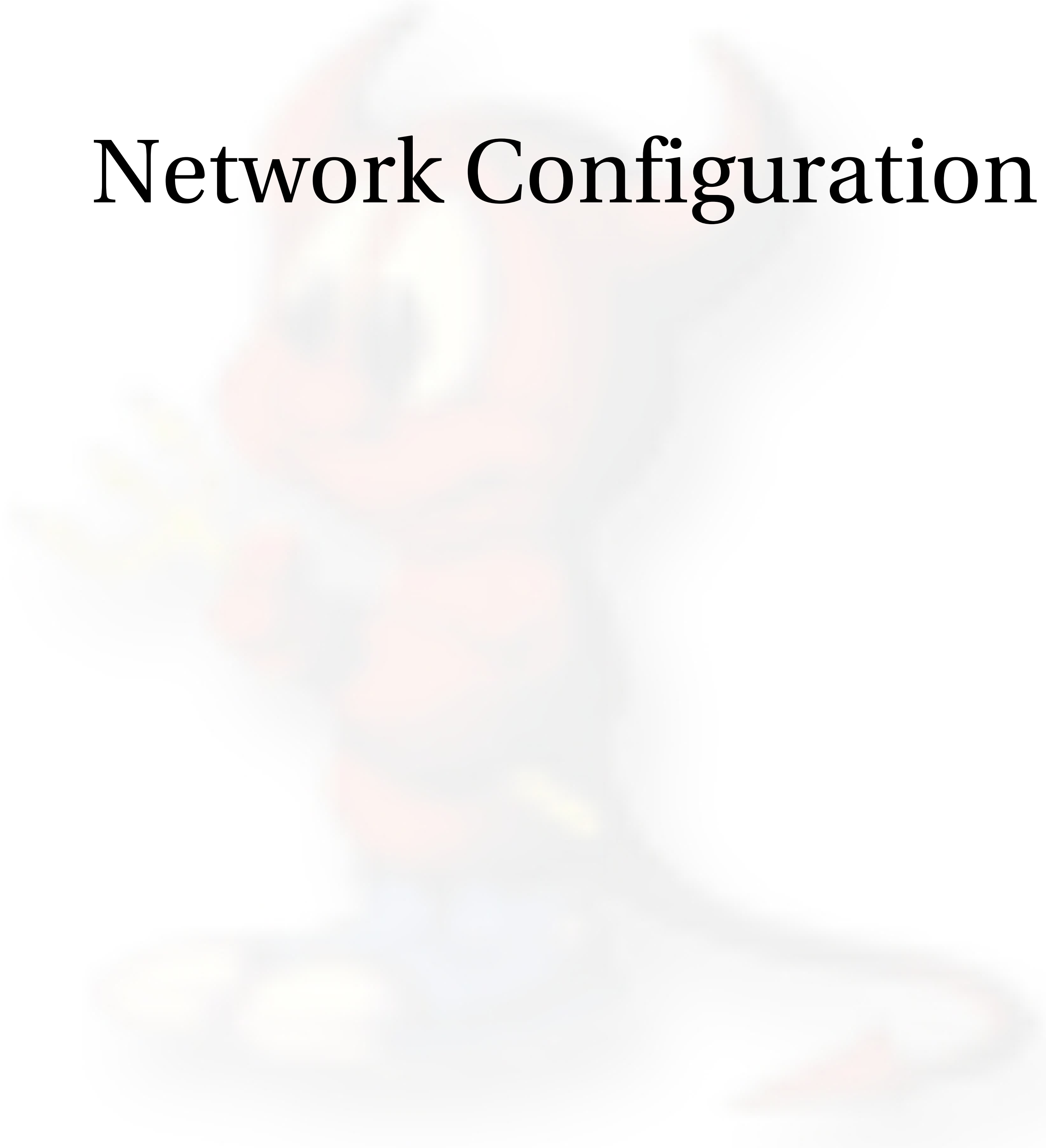
It makes installing and upgrading software *much* easier, and handles dependency problems.

Example: you want to upgrade Apache (remove it and reinstall a new version), but the system won't let you because PHP needs it.

-> Portupgrade handles this for you.

Portupgrade is itself a port and can be installed like any other port/package.

Network Configuration



Configuring Network Interfaces

During boot if a NIC is recognized then the appropriate code is loaded to support the NIC (a module).

After boot, using “`ifconfig`” you can see if the NIC exists. Look for MAC address.

Initial NIC configuration can be done with `ifconfig`, or try “`dhclient dev`”

If NIC works, edit /etc/rc.conf and put in device specific entries for each boot.

Configuring Network Interfaces cont.

Example lines in /etc/rc.conf for network device:

```
hostname="localhost.localdomain"
ifconfig_wi0="dhcp"          or
ifconfig_wi0="inet 192.168.0.100/24"
```

Set the hostname and indicate that NIC "`wi0`" uses DHCP or give IP address and netmask using "/" convention (/24=255.255.255.0).

Note that FreeBSD uses specific names for each network device. "`wi0`" indicates the first "Wireless" card.

Configuring Network Interfaces cont.

- FreeBSD 6 allows you to rename network interfaces as you like.
- Linux users who prefer “eth0” instead of “wi0” could configure this in /etc/rc.conf with:
  - `ifconfig_wi0 = “DHCP name eth0”`
- Some programs, however, expect specific-named network interfaces.

Summary

- Aimed at stability first, then user desktops.
- Very, very good track record for stability and security.
- Scales to very large sizes for services.
- Massive collection of software (16,800++ ports as of April 2007), including the ability to run Linux packages.
- Software can be installed in several ways.
  - FreeBSD `portupgrade` facility is superior to rpm as it can resolve dependencies.

More resources

This presentation is located here:

http://www.ws.afnog.org/bootcamp/intro-unix.pdf

- http://www.freebsd.org/
- http://www.freebsd.org/support.html
- O'Reilly books (http://www.oreilly.com/)
- http://www.freshports.org/
- http://www.freebsddiary.org/

Exercises 4

12 Basic network configuration
13Initial login, virtual terminals and command line