Chapter 10
System monitoring and logging

Chapter 10 Outline

- In this chapter we will learn how to:
  - ✓ Monitor system load
  - ✓ Monitor disk usage
  - ✓ Monitor log files

Monitoring system load

- Monitoring system load
  - The `/proc` file system
  - Examining files in `/proc`
  - Viewing system load with `xosview`
  - Viewing system load with KDE System Guard
  - KDE System Guard screenshot
  - Observing memory statistics with `vmstat`
  - `procinfo`
  - Other tools for summarising system load

The /proc file system

- The “files” under /proc make internal kernel information available via normal file read commands
  - There is a subdirectory for each process, named after the process ID

```
# ls -l /proc/1
total 0
dr-xr-xr-x    3 root     root    0 Apr 27 14:34 .
dr-xr-xr-x  112 root     root    0 Apr 27 08:38 ..
-r--r--r--    1 root     root    0 Apr 27 14:34 cmdline
lrwxrwxrwx    1 root     root    0 Apr 27 14:34 cwd -> /
-r--------    1 root     root    0 Apr 27 14:34 environ
lrwxrwxrwx    1 root     root    0 Apr 27 14:34 exe -> /bin/init
dr-x------    2 root     root    0 Apr 27 14:34 fd
-rw-------    1 root     root    0 Apr 27 14:34 mapped_base
-r--r--r--    1 root     root    0 Apr 27 14:34 maps
-rw-------    1 root     root    0 Apr 27 14:34 mem
-r--r--r--    1 root     root    0 Apr 27 14:34 mounts
lrwxrwxrwx    1 root     root    0 Apr 27 14:34 root -> /
-r--r--r--    1 root     root    0 Apr 27 14:34 stat
-r--r--r--    1 root     root    0 Apr 27 14:34 statm
-r--r--r--    1 root     root    0 Apr 27 14:34 status
```

The /proc file system (continued)

- The files in /proc can be examined with any standard command, e.g. cat
- This example shows the status of process ID=1 (init)
  - Name, PID, Parent PID
  - User and group identity
  - Virtual memory statistics
  - Signal status, etc ...
- This low-level data is used by tools such as top or KDE System Guard that display the information in a nicer format

```
# cat /proc/1/status
Name:   init
State:  S (sleeping)
Tgid:   1
Pid:    1
PPid:   0
TracerPid:      0
Uid:    0       0       0       0
Gid:    0       0       0       0
FDSize: 32
Groups:
VmSize:      620 kB
VmLck:         0 kB
VmRSS:       256 kB
VmData:      136 kB
VmStk:         8 kB
VmExe:       464 kB
VmLib:         0 kB
SigPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: fffffffd770d8fc
SigCgt: 00000000288b2603
CapInh: 0000000000000000
CapPrm: 00000000ffffffff
CapEff: 00000000fffffeff
```


Examining files in /proc

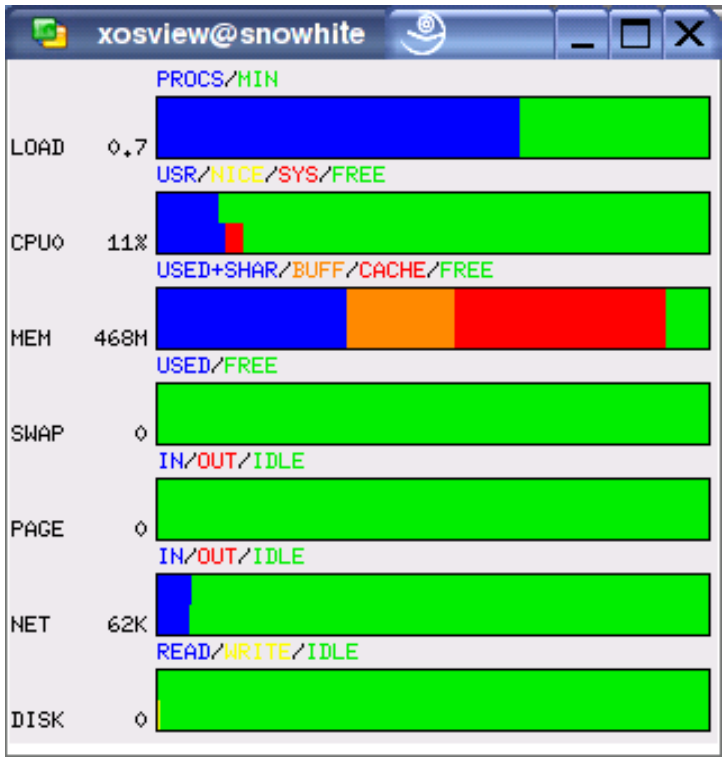
- Examine some of the files in /proc:

File	Description
/proc/cpuinfo	Information about the processor
/proc/interrupts	Allocation of interrupt vectors
/proc/ioports	Allocation of I/O port addresses
/proc/modules	Active kernel modules
/proc/partitions	The disk partitions and I/O statistics
/proc/mounts	Mounted file systems
/proc/fileystems	Supported file system formats
/proc/version	The kernel version
/proc/meminfo	Virtual memory statistics
/proc/sys/*	System and kernel information

Viewing system load with `xosview`

- `xosview` provides a convenient bar-graph display of system load
- Command line options add to or subtract from the set of bar graphs

```
$ xosview +disk -int
```



Upper bar shows instantaneous value
Lower bar shows averaged value

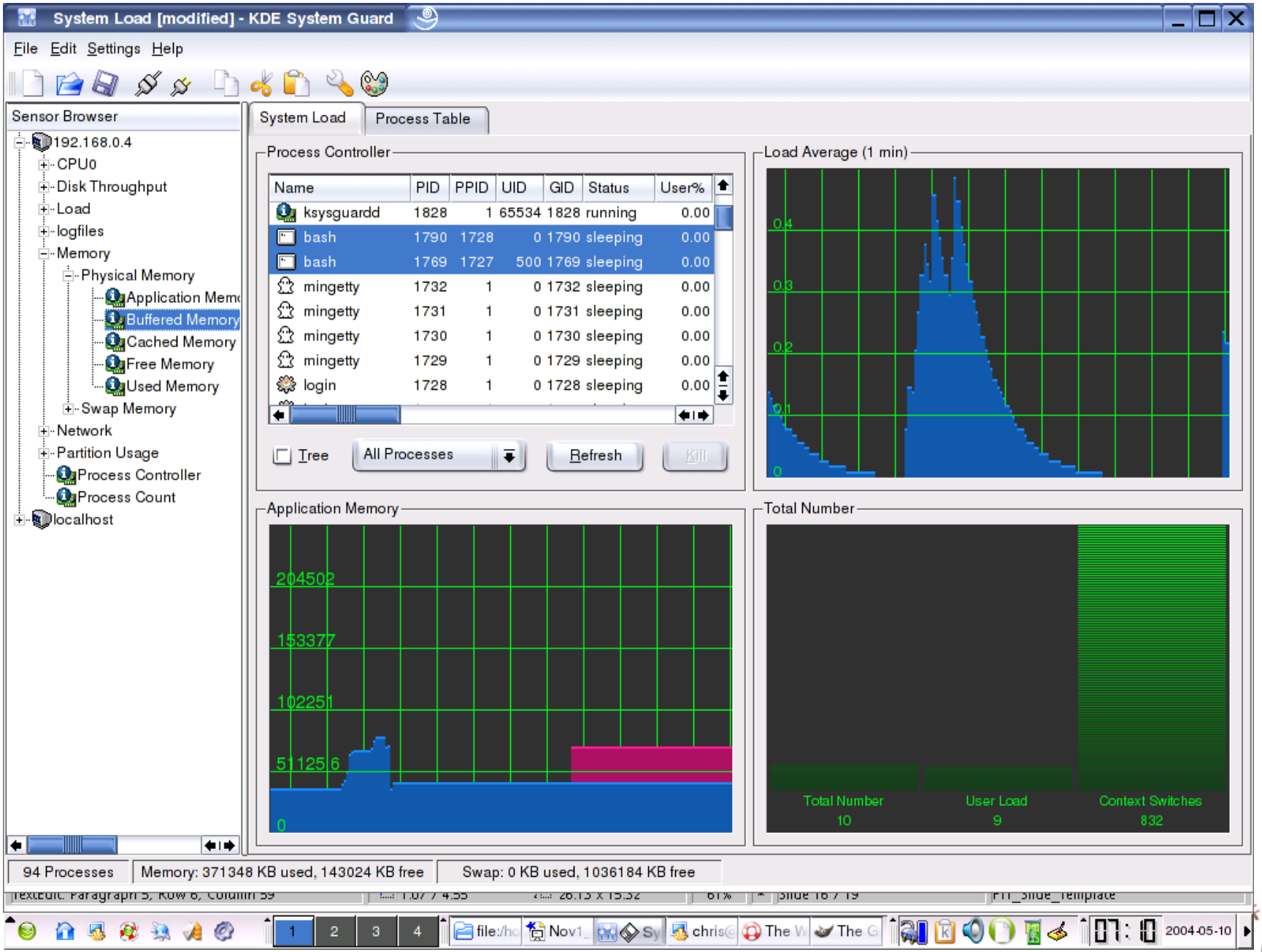
ict@innovation

FOSSFA

Viewing system load with KDE System Guard

- KDE system guard can chart the variation of system load over time
- *System load window* supports charting of a number of 'sensors'
  - Left-hand pane shows tree-structured view of sensors
    - CPU usage, system load, disk activity, network activity, memory usage, swapping, etc
  - Just drag and drop the sensors onto the charts
- Sensors can be displayed using several types of chart
  - Signal plotter: Scrolling chart showing variation over time
  - Multimeter: “Digital multimeter” display
  - Bar graph
  - Sensor Log: Log readings to file at specified interval

KDE System Guard screenshot



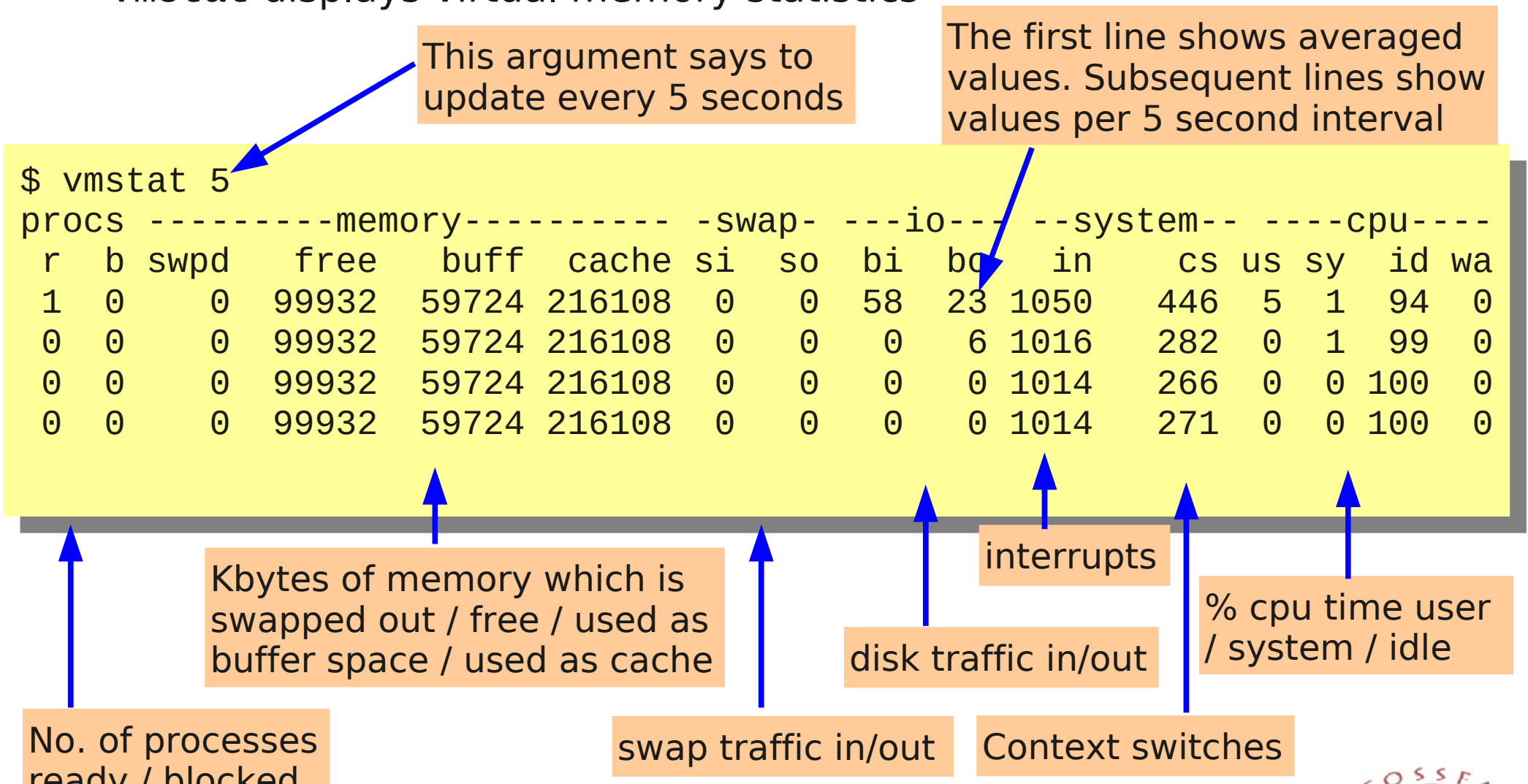
ict@innovation

Exercise: Using KDE System Guard

1. Start KDE system guard via the System -> Info -> KDE system guard menu
2. Select the “System Load” tab
3. Right-click on one of the four charts and select “Remove Display” from the pop-up menu
4. Expand the sensor browser view in the left-hand pane and locate the “Application Memory” sensor
5. Drag the sensor into the unused chart area and create a chart of type “Signal Plotter”
6. Drag the “Free Memory” sensor and add it to the chart
7. Remove one of the other four charts. Drag the “Process count” sensor into the unused chart area and create a chart of type “Multimeter”
8. Create and customise some other charts of your choice

Observing memory statistics with vmstat

- vmstat displays virtual memory statistics



ict@innovation

procinfo

- Another tool for summarising system load!

```
$ procinfo
Linux 2.4.21-99-default (root@i386.suse.de) (gcc 3.3.1 ) #1 1CPU [snowhite.]

Memory:      Total        Used        Free      Shared     Buffers      Cached
Mem:        514372      423056       91316           0       60480      218392
Swap:      1036184           0     1036184

Bootup: Tue May 11 14:52:26 2004    Load average: 0.14 0.10 0.09 1/98 2835

user  :       0:04:45.36   5.0%  page in :   229056  disk 1:    13882r    7462w
nice  :       0:00:00.00   0.0%  page out:   102313
system:       0:01:01.94   1.1%  swap in :        1
idle  :       1:30:03.11  94.0%  swap out:        0
uptime:       1:35:50.39         context :  2834201

irq  0:   5750407 timer                 irq  7:         1
irq  1:      2210 keyboard              irq  8:         2 rtc
irq  2:         0 cascade [4]           irq  9:       702 acpi
irq  3:         1                       irq 10:     11217 eth0, usb-uhci, O2 M
irq  4:         1                       irq 12:    197160 PS/2 Mouse
irq  5:         0 usb-uhci              irq 14:     22281 ide0
irq  6:         2                       irq 15:     55621 ide1
```


Other tools for summarising system load

- There are yet more tools for summarising system load
  - Keep in mind that most of these are simply presenting information read from the `/proc` file system
  - `uptime`: Report how long system had been up, and load average
  - `free`: Report virtual memory info
  - `netstat`: Reports a great deal of information about active network connections, routing tables, network traffic, etc.

Monitoring disk space

- Monitoring disk space
  - Showing free disk space with `df`
  - Disk usage reporting with `du`
  - Showing free disk space with `KDiskFree`
  - Showing disk usage with `KDirStat`

Showing free disk space with df

- df reports the used and free space on all mounted filesystems

```
$ df
Filesystem     1K-blocks      Used Available Use% Mounted on
/dev/hda3       10305376   5710432   4594944  56% /
tmpfs             257184         0    257184   0% /dev/shm
/dev/hda1        8192860   2103084   6089776  26% /mnt
/dev/sr0          246624    246624         0 100% /media/cdrom
```

- Options include:

Option	Meaning
-h	Show sizes in human-readable form (e.g. 234M, 2.2G) instead of in 1Kbyte blocks
-i	Show inode usage instead of disk block usage
-T	Show file system type

Disk usage reporting with du

- du reports the disk space (1K blocks) used by all directories within a specified directory

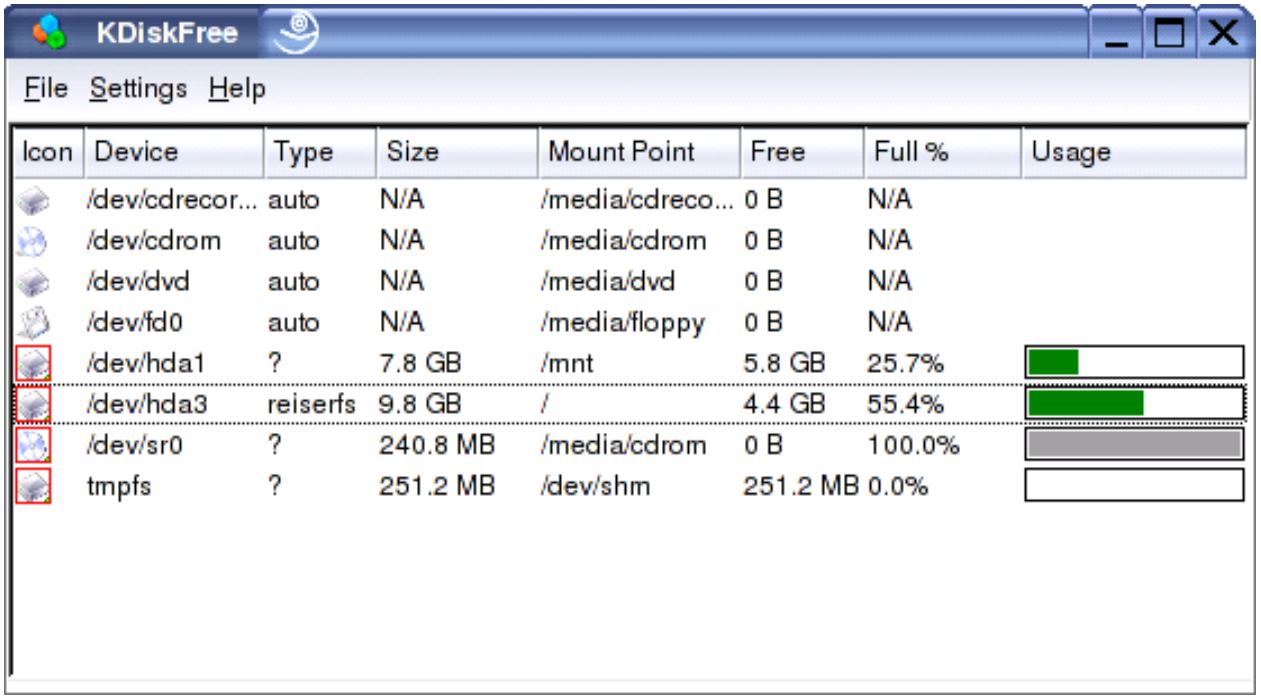
```
# du /lib | sort -nr | head -5
68079   /lib
59611   /lib/modules
59563   /lib/modules/2.4.21-99-default
42826   /lib/modules/2.4.21-99-default/kernel
31979   /lib/modules/2.4.21-99-default/kernel/drivers

# du -sh /lib
67M     /lib
```

- Options include:

Option	Meaning
-h	Show sizes in human-readable form (e.g. 234M 2.2G) instead of in 1Kbyte blocks
-a	Show counts for each individual file, not just directories
-s	Show only a total for each argument

Showing free disk space with KDiskFree

- KDiskFree (kdf) is a graphical version of df



ict@innovation

FOSSFA

Showing disk usage with KDirStat

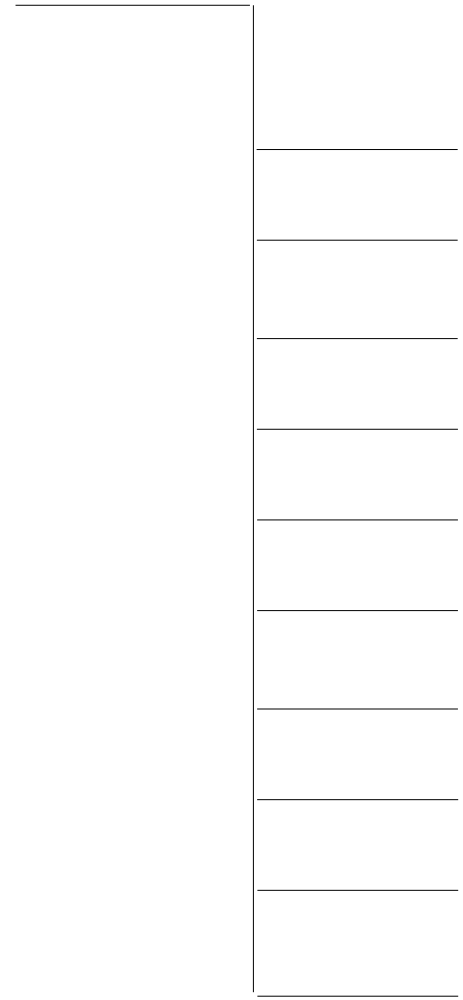
- KdirStat shows a tree structured view of disk usage along with a stunning graphical representation

ict@innovation

FOSSFA

Log files

- Log files



Log files

Key log files

The `syslogd` logging service

Facilities and levels for `syslogd`

Examples of rules in `syslog.conf`

Service-specific log files

Watching log files grow with `tail -f`

Log file analysis tools

Rotating log files

`logrotate` configuration file

Log files

- Many services report their activities by writing to log files
  - In conformance with the FHS standard, most log files are in `/var/log`
  - Some services can be configured to control the level of detail that they log
- Some log files are over-written each time the service starts
- Some log files are appended to and will grow without limit
  - Use `logrotate` to manage these
- Log files have several purposes
  - Investigate why services will not start up or operate correctly
  - Investigate abnormal conditions / configuration errors
  - Intrusion detection
  - Gather usage statistics (especially for web, ftp, mail services etc)

Key log files

Log file	Content
/var/log/boot.msg	Contains boot-time messages from the kernel and reports of service startup. It is overwritten each time the system is booted
/var/log/faillog	Contains failed login attempts. This is a binary file and should be examined with the command faillog
/var/log/warn	Various warning messages
/var/log/XFree86.0.log	Startup messages from the X server. Look in here if X will not start. Overwritten each time X is started
/var/log/XFree86.1.log	Ditto, but for display no. 1
/var/log/messages	The main system log file. The kernel and many services log to here
/var/log/samba/log.smbd	Log file for the samba file server
/var/log/samba/log.nmbd	Log file for the NETBIOS name server
/var/log/apache2/access_log	Access log for the apache web server
/var/log/apache2/error_log	Error log for the apache web server
/var/log/cups/access_log	Access log for the cups print server
/var/log/cups/error_log	Error log for the cups print server
/var/log/mail	Contains messages from the postfix mail delivery agent

The syslogd logging service

- Many services log messages through the daemon syslogd
- Messages are identified by
  - A facility: Which subsystem the message came from
  - A level: How important the message is
- The file /etc/syslog.conf determines what syslog will do with each message based on its facility and level. Messages may be:
  - Appended to a file
  - Written to a device
  - Forwarded to syslogd on some other machine. This is useful for consolidating logs on a single machine.
  - Written to a user
- Messages can be sent to more than one destination
- syslogd prepends a timestamp and the hostname to each message

Facilities and levels for syslogd

- Entries in `syslog.conf` consist of rules in the following format:

```
facility.level                                  action
```

One or more of:
```
auth
authpriv
cron
daemon
kern
lpr
mail
news
syslog
user
uucp
local0 - local7
```

One of:
```
debug
info
notice
warning
err
crit
alert
emerg
* = all
none
```

What to do with the message:

`/some/file`	append to file A leading '-' means do not force the file to be flushed to disk every time
`/dev/console`	write to device
`@hostname`	Send to `syslogd` on specified host
`username`	Write to specified user

Messages at *or above* the given priority will be logged

ict@innovation

Examples of rules in syslog.conf

- Here are some examples of rules in syslog.conf

```
*.crit                                  /var/log/warn
mail.*                                  -/var/log/mail
*.*;mail.none;news.none                 -/var/log/messages
*.*                                     @loghost
news.err                                -/var/log/news/news.err
*.alert                                 root
```

- If the configuration file is changed, syslogd can be made to re-read it by sending it a SIGHUP signal

```
# kill -SIGHUP $(cat /var/run/syslog.pid)
```

Exercise: Configuring syslogd

- Add a line to `syslogd`'s configuration file so that messages from the `local0` facility with priority `notice` are sent to the file `/var/log/notices`
- Send a `SIGHUP` signal to `syslogd` to force it to re-read the file
- The `logger` command can be used to log a message via `syslogd` from the command line (or from within a shell script)
- Execute the command:

```
$ logger -p local0.notice  "This is a test of syslogd"
```

- Examine the file `/var/log/notices`. Do you see the logged entry?
- Execute the `logger` command 4 more times (using command history)
- Re-examine `/var/log/notices`. How does `syslogd` handle consecutive identical messages?

Service-specific log files

- Some services (especially the more modern applications) write their own log files directly and do not use `syslogd`
- Examples:
  - Apache (`httpd`)
  - Samba (`smbd` and `nmbd`)
  - Print server (`cupsd`)
- The names of these log files are determined by the configuration files for the individual services
- Standard SuSE linux configurations conform to the Filesystem Hierarchy standard and put their log files under `/var/log`

Watching log files grow with `tail -f`

- The command `tail -f` is useful for monitoring log files as they grow
  - Shows tail of file then waits and shows any lines appended to the file
- Open two terminal windows

  - In one window, as root, run `tail -f /var/log/mail`
  - In the other window, as tux, send a mail message:

```
$ mail -s testing tux
this is just a test
^D
```

- Look at the output from `tail -f`
  - How many lines were appended to the file? _____
  - Which service wrote them? _____
- Enter `^C` to quit from `tail`

Log file analysis tools

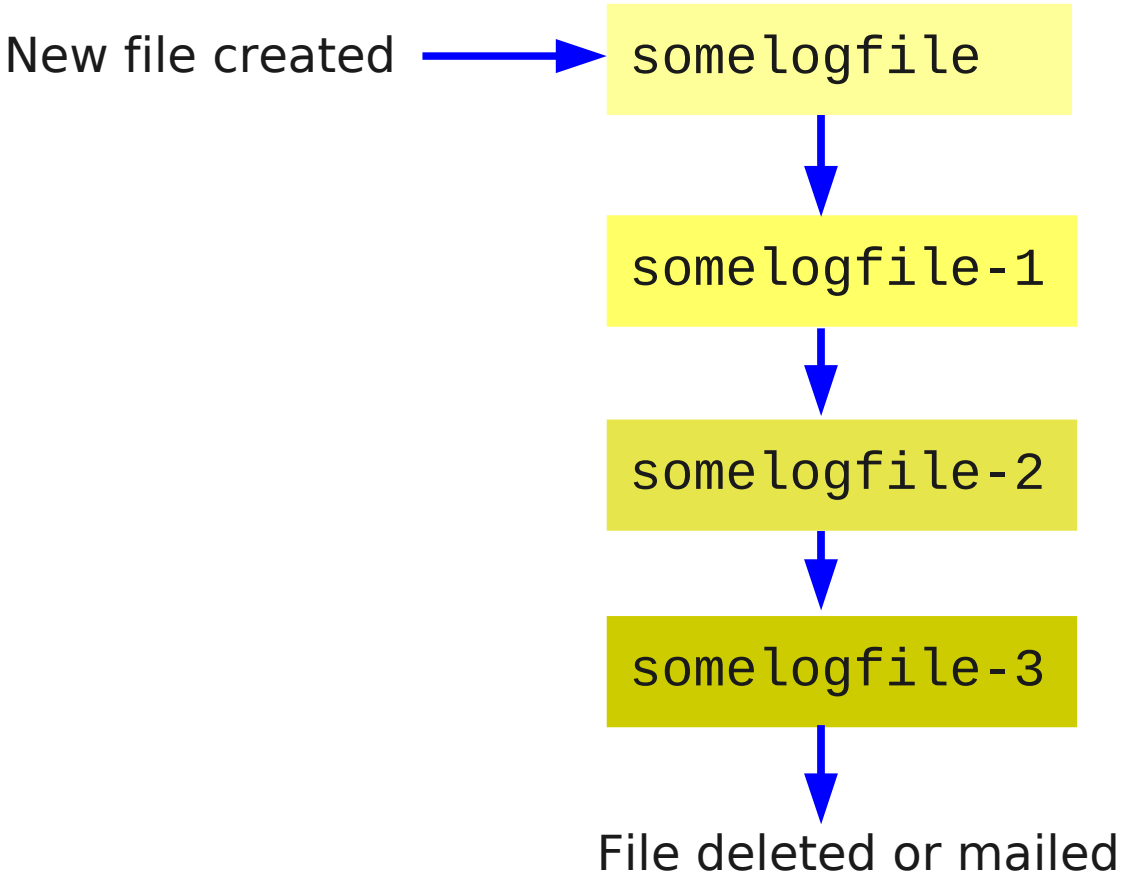
- Administrators prefer not to read log files routinely
- Tools exist to monitor log files automatically
  - Usually configured to look for regular expression matches within the file
  - Can generate mail or otherwise alert the system administrator
- Examples: (none of these ships with SuSE linux)
  - swatch One of the earliest tools; written in perl
  - logsurfer A later tool; see `www.cert.dfn.de/eng/logsurf`
  - logrep Monitors and consolidates log files and generates HTML formatted reports. See `logrep.sourceforge.net`
  - analog, wwwstat, wusage: Tools for summarising apache access logs
  - sawmill A commercial (i.e. not free) tool for summarising logs, especially web traffic. See `www.sawmill.net`

Rotating log files

- Without intervention, many log files will grow without bound
  - Much of the data is too old to be useful
- *Rotation* of logs divides them into files of manageable size and keeps only the most recent
- The utility `logrotate` automates this process
  - Automatic rotation, compression, removal, and mailing of log files.
  - Each log may be rotated daily, weekly, monthly, or when it grows too large.
- Normally, `logrotate` is run as a daily cron job
- Has its own configuration file specifying which log files to manage
  - On SuSE linux this file is `/etc/logrotate.conf`; in turn this file includes config files for the individual services in the directory `/etc/logrotate.d`

Rotating log files (continued)

- The existing log file is renamed with '-1' on the end
  - The '-1' file is renamed '-2', and so on
  - The oldest file is discarded or mailed



logrotate configuration file

```
# sample logrotate configuration file
        compress

        /var/log/messages {
            rotate 5
            weekly
            postrotate
                /sbin/killall -HUP syslogd
            endscript
        }

        /var/log/news/news.crit {
            size 100k
            rotate 2
            olddir /var/log/news/old
            missingok
            postrotate
                 kill -HUP `cat /var/run/inn
            endscript
            nocompress
        }
```

Name of log file to manage

Number of old files to keep

How often to rotate the log
Also daily, monthly

Command(s) to execute after the log has been rotated

Rotate if log is bigger than this

Where to put old log files

Do not report an error if the log file is missing

Do not compress the old logs

Quiz

- Name three tools for displaying system load information
- What command would you use to discover the cache size of the CPU?
- What command would you use to display the total number of disk blocks used by the files in `/etc`?
- This line shows the column headings of which command?

```
Filesystem     1K-blocks      Used Available Use% Mounted on
```

- Which file is this line taken from?

```
*.*;mail.none;news.none        -/var/log/messages
```

- True or false?
  - The files in `/proc` are stored in the root partition
  - Apache writes its log file directly, it does not use `syslogd`