Debian/GNU Linux

Working on a Command Line

Károly Erdei

November 8, 2013

Agenda

Working on a command line

i.e. working in a shell

Summary

- you work in a terminal window on the console
- starting the terminal window:
  - Gnome: Applications: Accesoirs: Terminal / Konsole
- more terminal programs are available
  - konsole (KDE), gnome-terminal (gnome), xterm the original X terminal
- connecting remotely to a terminal window (to a computer)
  - from a MS Windows computer by SSH (putty, www.putty.org)
  - from other Linux/Unix/Mac computer: ssh -X -l username computername
- you are working with a shell in the terminal window

Working on a command line

i.e. working in a shell

Advantages

- common for every Linux/Unix/Mac OS-X system
- more flexible than a windowing interface
- based on commands you can write (big) shell programs
- according your rights
  - you can start (all) programs on the system
  - you have (full) control on the OS
  - as root user you have the full control without restrictions by GUIs

Most important advantage

- using the command you have full access to all parameters
- no restictions against GUI represented solution
  - using GUI only access to implemented features is available !!!

Working on a command line

Useful commands: an overview - I

File system

- Manage files:
  - touch (create file); rm (delete file); cat (list file); cp (copy file); mv (move file); cmp (compare files);
- Manage directories:
  - mkdir (create directory); rmdir (remove empty directory); rm -r (remove directory); mv (move directory); cp -r (copy directory); pwd (print working directory); cd (change directory); ls (list directory content);
- Manage attributes:
  - chmod (change attributes), chown (change owner), chgrp (change group), umask;
- Other commands:
  - du (disk usage), df (disk free), ln (create symbolic link), mount (mount disk partition)

Working on a command line

Useful commands: an overview - II

Processes

- kill, killall, nice, ps, sleep, top

User environment

- finger, id, passwd, su, sudo, uptime, w, wall, who, whoami, talk

Text processing

- awk, cut, join, ed, sed, head, tail, less, more, sort, strings, vi, wc

Networking

- netstat, ping, traceroute

Searching

- find, locate, which; grep

Packaging

- dpkg

Working on a command line

Useful commands: The Top Ten

Useful commands:

- man man; man top; man ps;
- ls -laRtrF (d: directory; -:ordinary files; l: link)
- top - display Linux tasks; htop
- ps - list processes
- kill - kill processes
- find . -name "*pattern*"
  - man find, please check, extreme powerful command
  - find /tmp -name EXP; find / -name "*latex*"
- du -s *
- grep -r -i -v pattern path; egrep; searchmonkey
  - grep process wheezy-allpackages.txt | wc
  - grep " processes" software/wheezy-allpackages.txt | wc
- do NOT forget the difference between OS commands and shell builtin commands !

Agenda

Working on a command line

How to get information about commands, files, etc.

Man pages

- if command not known:
  - man -k topic (e.g.: man -k file; man -k shell; man -k ls)
- structure of man pages (learn, check: 8 sections)
  - man man; man tty; man 1 tty; man 4 tty;
- xman: graphical tool

Info pages

- another structuring of the information
- man info; info info, etc. (often man page points to info page)

T LDP - The Linux Documentation Project

- http://www.tldp.org/
- HOWTOs, GUIDEs, FAQs, Wiki, etc.
- http://man.sourcentral.org/debian-squeeze/ (lot of languages)

Working on a command line

examples: man -k

```
hades:sysadmin!16> man -k "file manager"
filerunner (1)         - simple and efficient file manager with FTP
fr (1)                 - simple and efficient file manager with FTP
gnome-commander (1)    - A GNOME file manager
konqueror (1)          - Web browser, file manager, ...
nautilus (1)           - the GNOME File Manager
hades:sysadmin!17>

hades:sysadmin!29> man -k "ripper"
grip (1)               - A gtk-based cd-player and cd-ripper
kaudiocreator (1)      - CD ripper and audio encoder front-end
sound-juicer (1)       - GNOME-desktop CD ripper and player using GStream
hades:sysadmin!30>
```

Agenda

Working on a command line

Usefull commands: ls

ls -laRtrFd

- l: use a long listing format
  - first character in the list entry:
  - d: directory; -:ordinary files; l: link;
- a: list hidden files (do not ignore entries starting with .)
- R: list subdirectories recursively
- t: sort by modification time
- r: reverse order while sorting
- d: list directory entries instead of contents
- 1: list one file per line
- F: append indicator (one of */=>@|) to entries
- and a lot of other parameters

Agenda

Working on a command line

Usefull commands: top

top - display Linux tasks

```
top - 10:34:51 up 3 days,  5:15,  2 users,  load average: 0.09, 0.06, 0
Tasks: 125 total,   2 running, 123 sleeping,   0 stopped,   0 zombie
Cpu(s):  2.2%us,  0.2%sy,  0.0%ni, 97.7%id,  0.0%wa,  0.0%hi,  0.0%si,
Mem:   1034352k total,  1002484k used,    31868k free,   228300k buffer
Swap:  3028212k total,    51428k used,  2976784k free,   260068k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 9840 craab     20   0  179m  98m  17m S    5  9.7 104:16.82 Mathematic
 8123 craab     20   0 89908 6040 2884 S    1  0.6  14:14.38 gnome-term
18956 root      20   0  2888  688  540 S    1  0.1   0:01.06 dirmngr
26613 sysadmin  20   0  2356 1168  920 R    1  0.1   0:00.12 top
 8117 craab     20   0 35348  27m 2888 S    0  2.7   7:40.05 Xvnc4
 9847 craab     20   0  223m  64m  12m S    0  6.4  17:37.62 MathKernel
 9990 craab     20   0  435m  33m 8748 S    0  3.3  15:14.87 java
```

Working on a command line

top - most important information

load avarage

- system load state

Cpu usage:

- us, sy, id

Memory

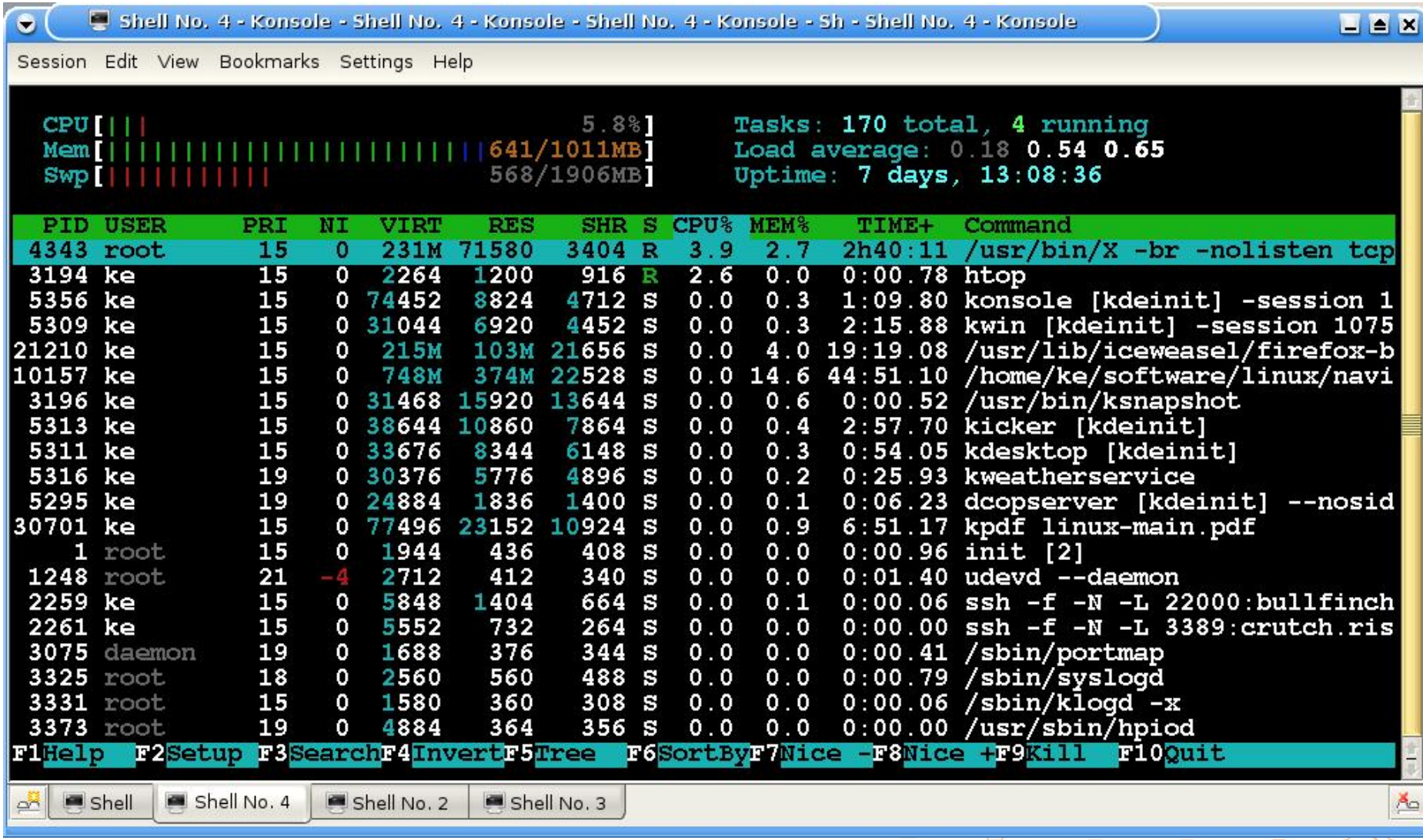
- used / free

Swap

- used

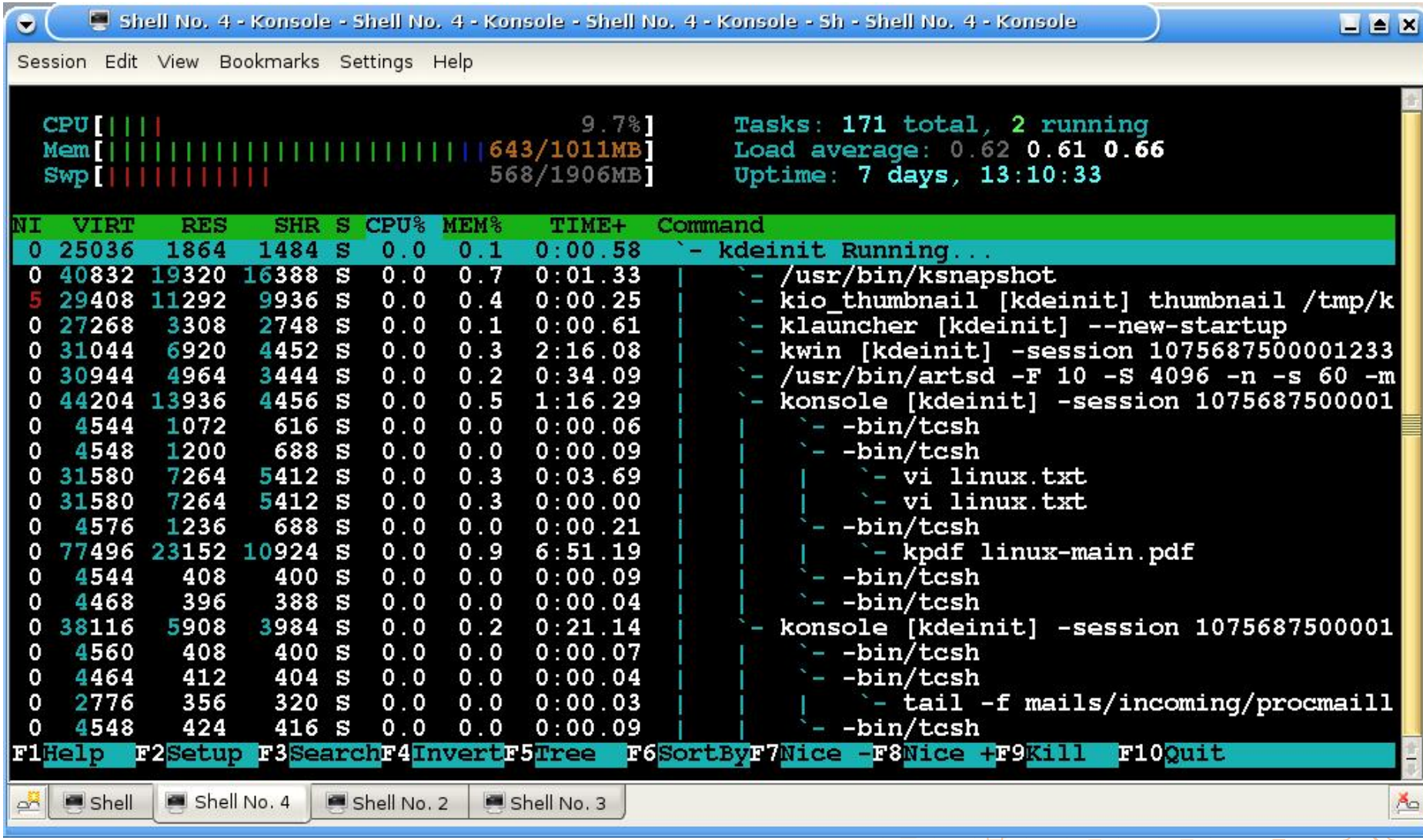
Working on a command line

Usefull commands: htop - basic view



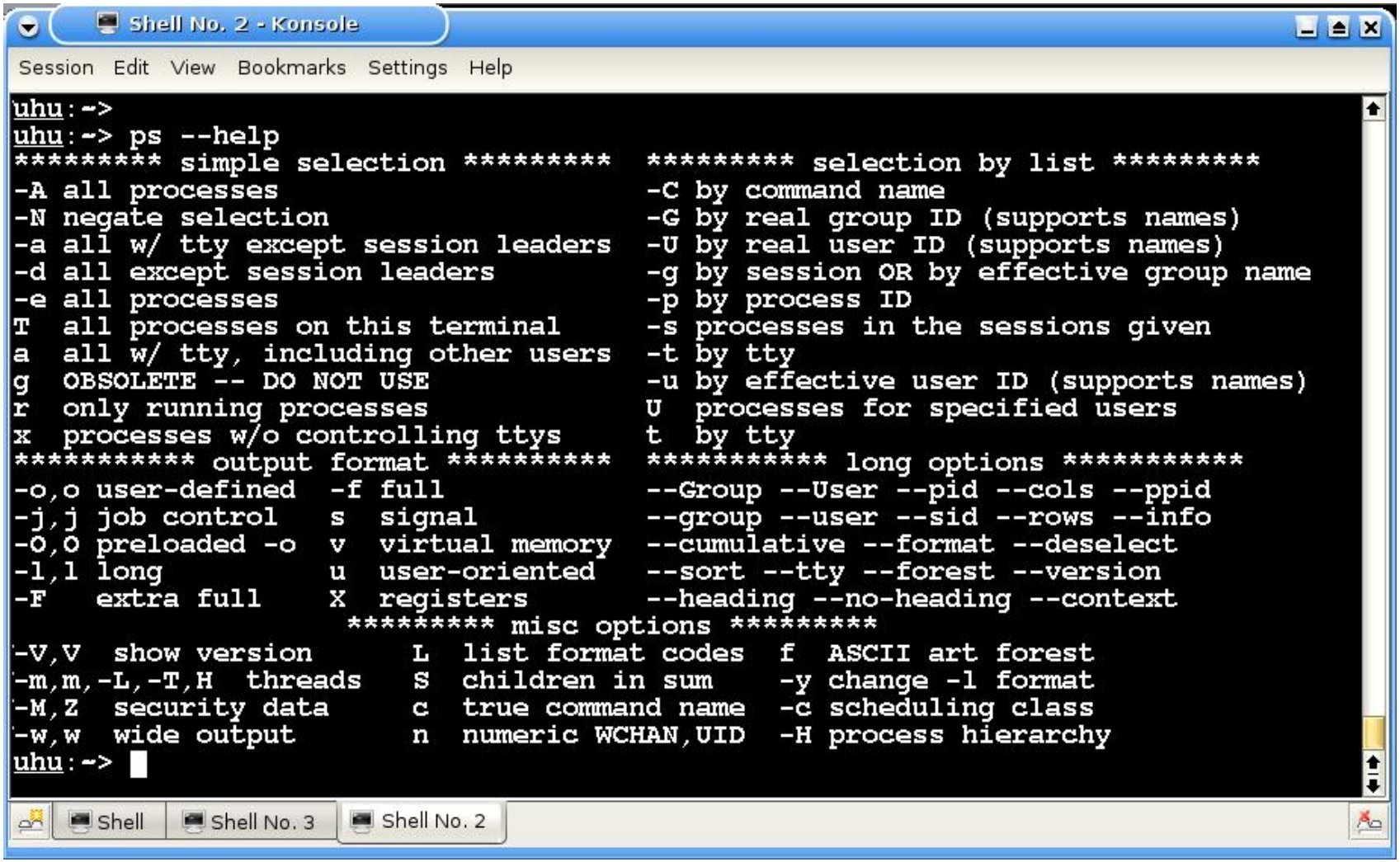
Working on a command line

Usefull commands: htop - tree view of processes



Working on a command line

Usefull commands: ps - the help page of ps



Working on a command line

Usefull commands: ps - an example

ps auxw: To see every process on the system (BSD like format)

```
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAN
craab      9840  2.4  9.9 186784 103100 pts/1   Sl   Oct29 110:17
     /zvol/mathematica/mathematica-7.0/SystemFiles/FrontEnd/Binaries/Li
root       4197  0.0  0.0   8736   828 ?        Ss   Oct29   0:05
sendmail: MTA: accepting connections
craab     14143  0.2  0.1   2480  1204 pts/1    S+   Oct29  10:36 top
```

ps auxw | grep sysadmin

```
root      27329  0.0  0.2   9372  2972 ?        Ss   12:35   0:00 sshd:
sysadmin 27336  0.0  0.1   9372  1612 ?        S    12:35   0:00 sshd:
sysadmin 27337  0.0  0.2   4644  2328 pts/2    Ss   12:35   0:00 -tcsh
sysadmin 27494  0.0  0.0   2396   952 pts/2    R+   12:50   0:00 ps aux
sysadmin 27495  0.0  0.0   1832   540 pts/2    S+   12:50   0:00 grep s
```

Working on a command line

Usefull commands: kill

Terminate processes: kill, killall (/bin/kill)

- kill -TERM proc-ID (sends the terminate signal)
  - -TERM is default
- kill -KILL proc-ID (sends the KILL signat)
  - this signal may not be blocked
  - KILL or 9 (numeric values available)
- /bin/kill is the path for this programm

shell builtin command: kill [-s signal] job | pid

- signals are identical of the /bin/kill program
- you can kill jobs, too.
- default signal is TERM (terminate)
- examples:
  - kill %1; kill PID; kill -9 PID

Working on a command line

Usefull commands: kill - how to kill jobs

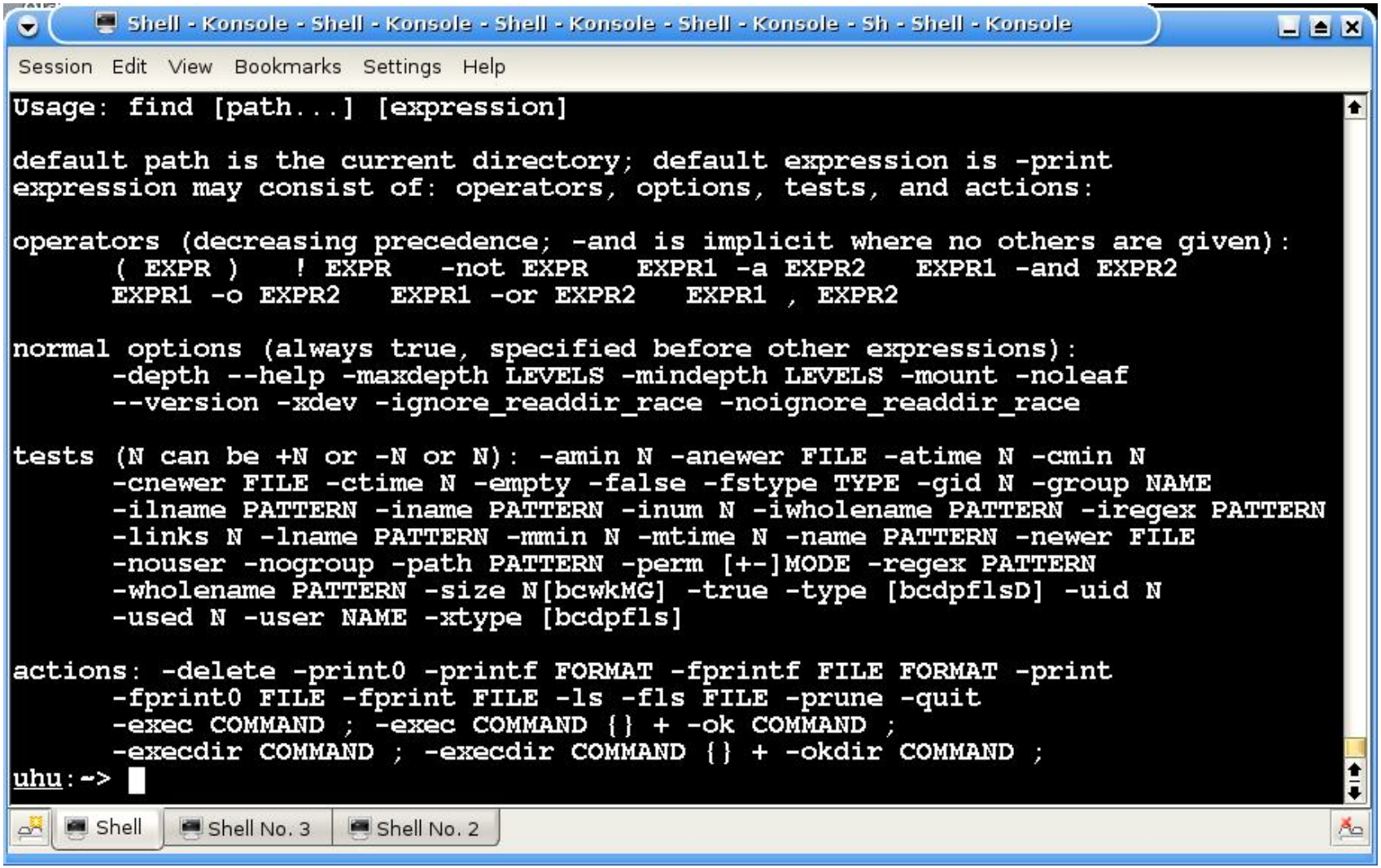
```
uhu:~/cbwe-2009/linux> kpdf linux-main.pdf &
[1] 14338
uhu:~/cbwe-2009/linux>
uhu:~/cbwe-2009/linux> kpdf  ~/CBWE-certificatespdfpreview.pdf

Suspended
uhu:~/cbwe-2009/linux> bg
[2]    kpdf ~/CBWE-certificatespdfpreview.pdf &
uhu:~/cbwe-2009/linux>
uhu:~/cbwe-2009/linux> jobs
[1]  + Running                 kpdf linux-main.pdf
[2]    Running                 kpdf ~/CBWE-certificatespdfpreview
uhu:~/cbwe-2009/linux>
uhu:~/cbwe-2009/linux> kill %2
uhu:~/cbwe-2009/linux>
[2]    Terminated              kpdf ~/CBWE-certificatespdfpreview
uhu:~/cbwe-2009/linux>
```

Agenda

Working on a command line

Usefull commands: find - the help page



Working on a command line

Usefull commands: find

find command

- By default, find returns all files below the current directory
- allows user to specify an action to be taken on each matched file
- extremely powerful program for applying actions to many files

find [-H] [-L] [-P] [path...] [expression]

- At least one path must precede the expression
- capable of interpreting wildcards internally
- commands must be constructed carefully in order to control shell globbing
- supports regex matching; expression elements are whitespace-separated and evaluated from left to right
- by default, find executes the '-print' action
- The GNU find has a large number of additional features not specified by POSIX.

Working on a command line

Usefull commands: examples of find

find . -name "*mp3"

```
./mp3-what/csendes.mp3
./.gnome/Gnomp3
./private/DanielGerard-Butterfly.mp3
```

- searches current directory (represented by a period) and below it
- the quotes avoid the shell expansion

find . -size 15M

```
./skype-debian_2.0.0.72-1_i386.deb
./mails/mails-lists/2006/greifvoegel-1
./mails/spam-last-part
./nature-docs/at/arge-orni/OOe-BVAtlas-methode.pdf
```

Working on a command line

Usefull commands: examples of find

```
find . -size 15M -exec ls -l {}  \;

-rw-r--r-- 1 ke ke 15504764 2009-08-27 10:45 ./skype-debian_2.0.0.72-1_
-rw-rw-r-- 1 ke ke 15037423 2006-12-31 23:32 ./mails/mails-lists/2006/g
-rw-r--r-- 1 ke ke 14970181 2007-03-30 10:14 ./mails/spam-last-part
-rw------- 1 ke ke 14965394 2005-06-14 11:41 ./nature-docs/at/arge-orni
```

parameter

- {} means: replacing {} with the name of the file
- the semicolon indicates the end of the command
  - backslashed to avoid the shell interpreting it as a command separator

find

```
find /scratch/ke -name "*.mp3" -type f -exec chmod 400 {} \;
```

Working on a command line

Usefull commands: examples of find

```
uhu:~/cbwe-2009/linux> find . -name "*toc" -print
./linux-2-main.toc
./linux-main.toc
uhu:~/cbwe-2009/linux>

uhu:~/cbwe-2009/linux> find . -name "*toc" -exec ls -l {} \;
-rw-r--r-- 1 ke ke 267 2009-11-02 12:03 ./linux-2-main.toc
-rw-r--r-- 1 ke ke 388 2009-11-13 20:41 ./linux-main.toc
uhu:~/cbwe-2009/linux>

uhu:~/cbwe-2009/linux> find . -name "*toc" -exec  ls {} \;
./linux-2-main.toc
./linux-main.toc
uhu:~/cbwe-2009/linux>
```

Agenda

Working on a command line

Usefull commands: du

du: Summarize disk usage of each file, recursively for directories

- du . (for the current directory)

```
1948    ./documents/misc
1572    ./documents/bva-docs
38524   ./documents/books/from-GG
2820    ./documents/books/linux
120     ./documents/books/digi-foto/macro_files
```

- du -s path
  - display only a total for each argument (., *, name)

```
uhu:~> du -s documents/
481452  documents/
uhu:~>
```

- du -s .??*
  - display total for the hidden dot-directories

Working on a command line

Usefull commands: df - report file system disk space usage

```
hades:sysadmin!5> df
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/hda1              14421344  11933544   1755240  88% /
tmpfs                    513564         0    513564   0% /lib/init/rw
udev                      10240       716      9524   7% /dev
tmpfs                    513564         0    513564   0% /dev/shm
/dev/hda5              22643764  20763036    730476  97% /zlocal/hda5
atlantis:/zlocal/sdb7/sysadmin
                       15377920  12132608   2464256  84% /home/sysadmin
atlantis:/zlocal/sda8/maple
                       15377920  10851840   3745024  75% /zvol/maple
atlantis:/zlocal/sda6/mathematica
                       15377920  11300864   3296000  78% /zvol/mathemati
hades:sysadmin!6>
```

Working on a command line

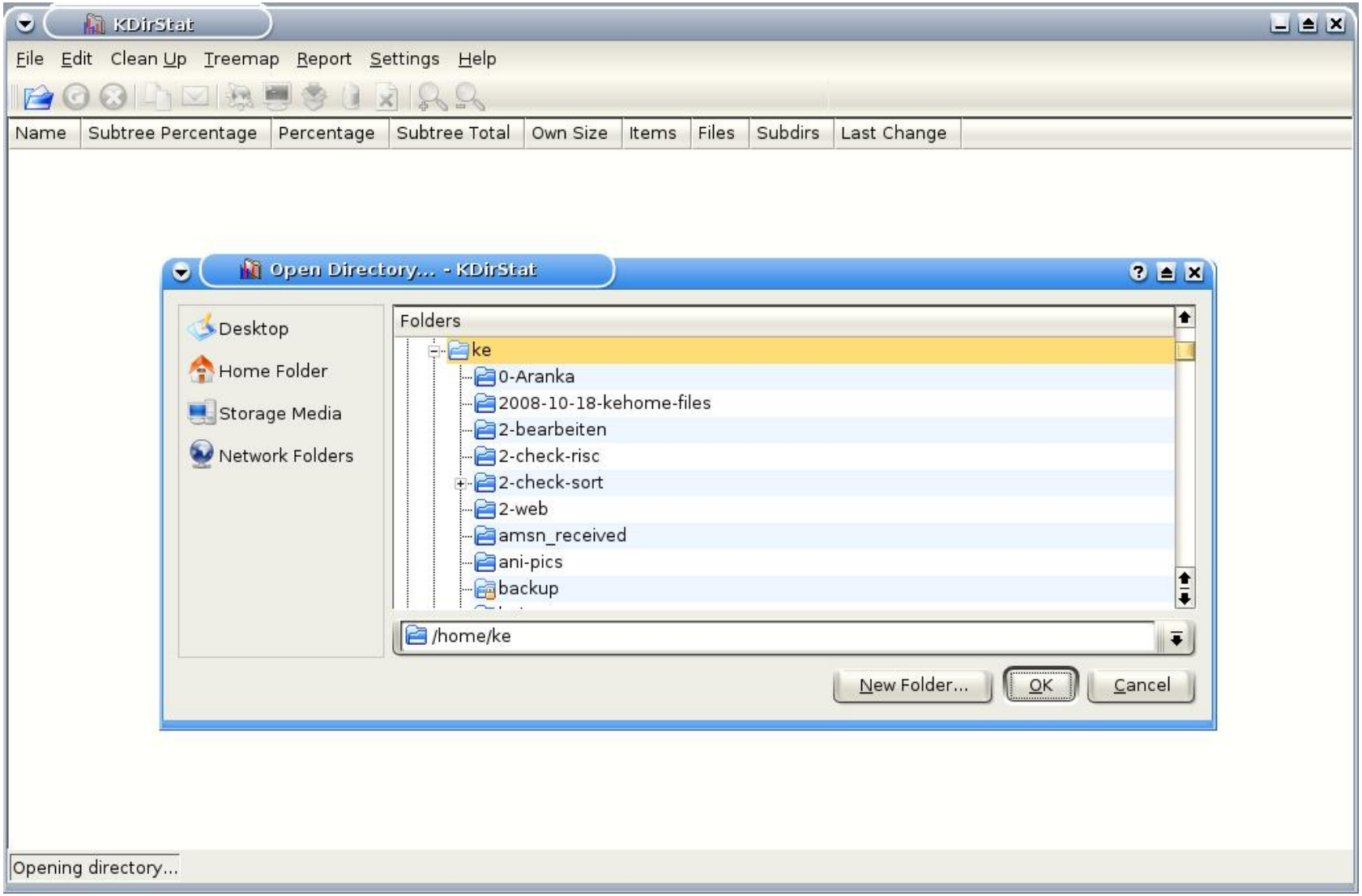
Usefull commands: examples of df

```
hades:sysadmin!6> df -l
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda1             14421344  11933544   1755240  88% /
tmpfs                   513564         0    513564   0% /lib/init/rw
udev                     10240       716      9524   7% /dev
tmpfs                   513564         0    513564   0% /dev/shm
/dev/hda5             22643764  20763036    730476  97% /zlocal/hda5

hades:sysadmin!7> df -l -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda1              14G   12G  1.7G  88% /
tmpfs                 502M     0  502M   0% /lib/init/rw
udev                   10M  716K  9.4M   7% /dev
tmpfs                 502M     0  502M   0% /dev/shm
/dev/hda5              22G   20G  714M  97% /zlocal/hda5
hades:sysadmin!8>
```

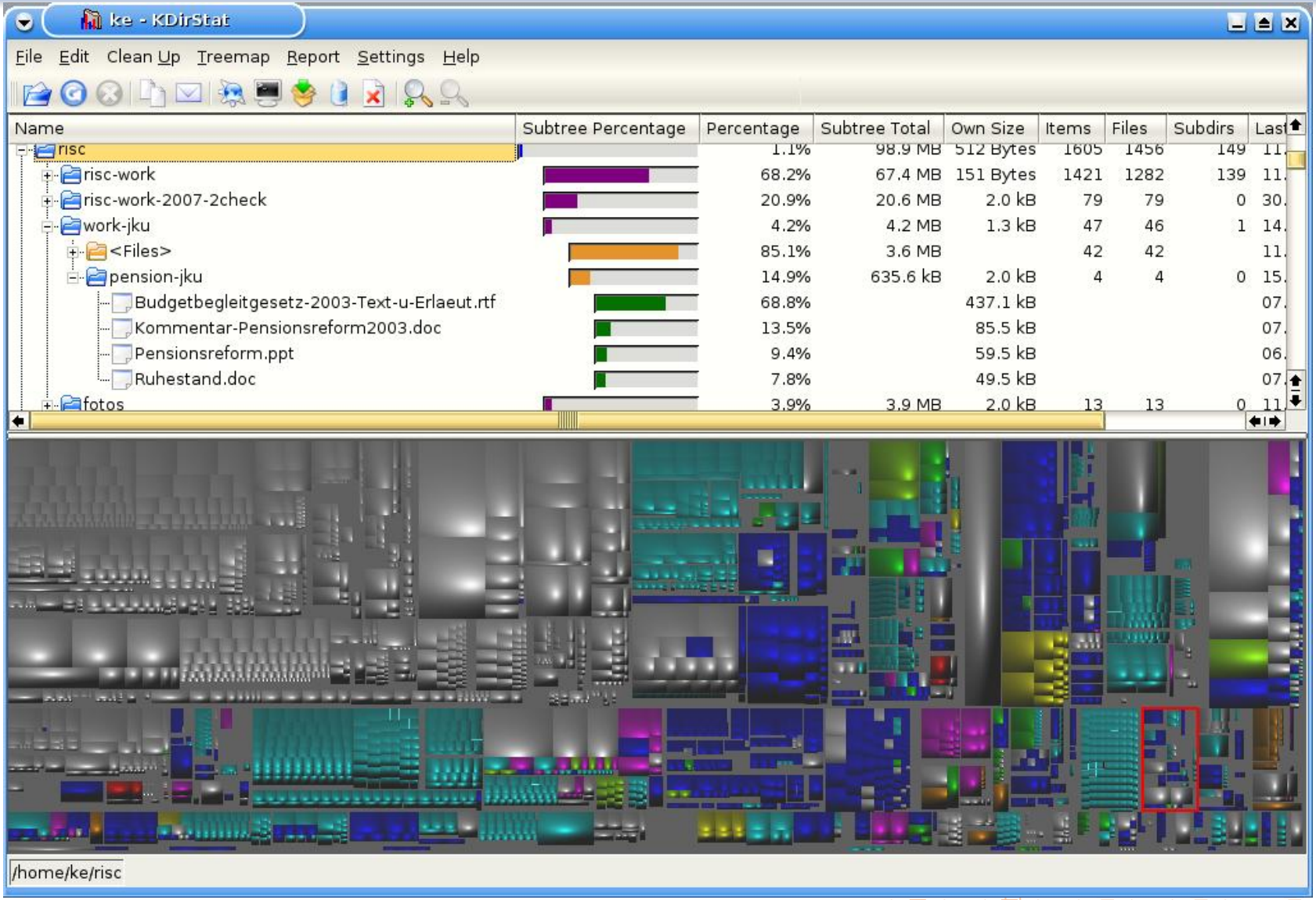
Working on a command line

Kdirstat - graphical view of the disk usage - the start screen



Working on a command line

Kdirstat



Working on a command line

Kdirstat

Agenda

Working on a command line

Commands in the user environment

Commands: uptime and who

- uptime - Tell how long the system has been running

```
uhu:~> uptime
 19:51:40 up 3 days,  7:33,  2 users,  load average: 0.11, 0.16, 0.
uhu:~>
```

- who - show who is logged on

```
gonzales:sysadmin!2> who
cschneid pts/1        Nov 18 11:58 (ozelot.risc.uni-linz.ac.at)
cschneid pts/0        Nov 18 12:06 (ozelot.risc.uni-linz.ac.at)
cschneid pts/2        Nov 18 12:11 (ozelot.risc.uni-linz.ac.at)
cschneid pts/3        Nov 19 15:33 (ozelot.risc.uni-linz.ac.at)
cschneid pts/4        Nov 18 13:52 (ozelot.risc.uni-linz.ac.at)
cdoench  pts/5        Nov 20 09:50 (dog.risc.uni-linz.ac.at)
mkauers  pts/6        Nov 21 12:01 (fennek.risc.uni-linz.ac.at)
fstan    pts/7        Nov 22 11:30 (ap164144.wlan.jku.at)
cschneid pts/8        Nov 22 15:21 (ozelot.risc.uni-linz.ac.at)
sysadmin pts/9        Nov 22 19:54 (hades.risc.uni-linz.ac.at)
gonzales:sysadmin!3>
```

Working on a command line

Commands in the user environment

w - Show who is logged on and what they are doing

```
gonzales:sysadmin!1> w
 19:54:20 up 4 days,  8:03, 10 users,  load average: 4.78, 4.37, 4.18
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
cschneid pts/1    ozelot.risc.uni- Wed11   32:45   2:54m  3.96s top
cschneid pts/0    ozelot.risc.uni- Wed12   24:29m 13:44   0.06s -tcsh
cschneid pts/2    ozelot.risc.uni- Wed12   24:32m 33:40   0.12s -tcsh
cschneid pts/3    ozelot.risc.uni- Thu15    2:23m  4:35m  0.32s tail -f
cschneid pts/4    ozelot.risc.uni- Wed13   24:27m  4:41m  0.68s -tcsh
cdoench  pts/5    dog.risc.uni-lin Fri09    2days  0.20s  0.20s -tcsh
mkauers  pts/6    fennek.risc.uni- Sat12   31:52m  1:25   1:24  top
fstan    pts/7    ap164144.wlan.jk 11:30   50:48  50.24s 50.22s top
cschneid pts/8    ozelot.risc.uni- 15:21    4:16m  4:17m  0.16s -tcsh
sysadmin pts/9    hades.risc.uni-l 19:54    0.00s  0.02s  0.00s w
gonzales:sysadmin!2>
```

Agenda

Working on a command line

Installing Debian packages

Debian package system

- package structure: main contrib non-free
  - (file name of package ends: .deb)
- http://www.at.debian.org/distrib/packages/
- http://packages.debian.org/stable/
  - get the file: (compact compressed textlist) - allpackages.htm
  - search it with grep by keywords for topics

Installing Debian packages (command line)

- apt-get install package-name (as root !)
- use the aptitude command (for advanced user)
  - sudo aptitude install package-name (will use a graphical terminal)
- sudo apt-get install searchmonkey

Working on a command line

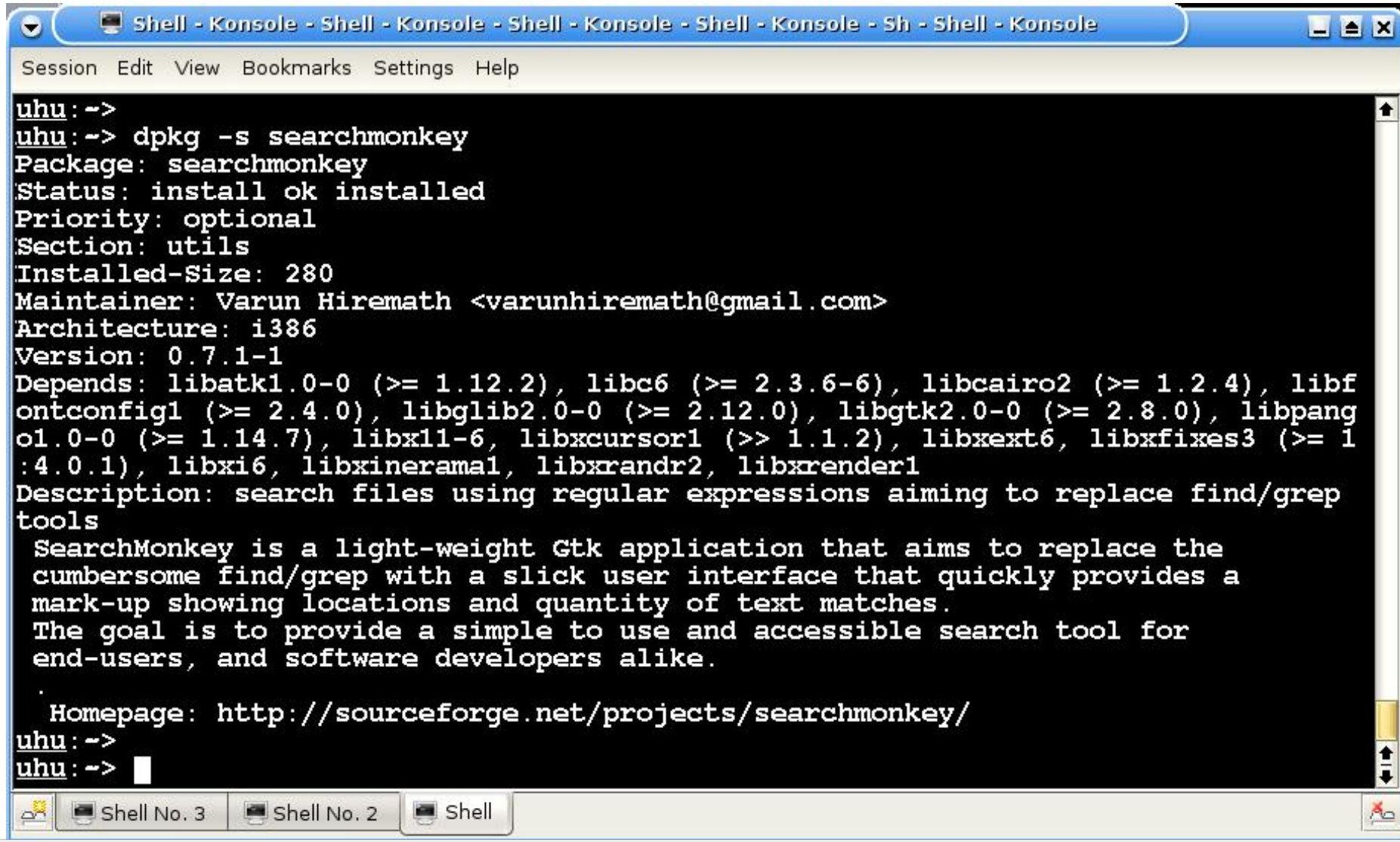
Installing Debian packages - dpkg - options summary

Using dpkg

- Install a Debian package: -i
  - sudo dpkg -i skype-debian_2.1.0.47-1_i386.deb
  - You can install any Debian package using: sudo dpkg -i ... on any RISC computer !
- Search for package name: -s
  - dpkg -s searchmonkey
- Search for file name in all Debian packages: -S
  - dpkg -S /usr/bin/searchmonkey
- List the contents of a Debian package : -L
  - dpkg -L searchmonkey
- List package names: -l
  - dpkg -l "browser*"
- LIVE presentation of dpkg !

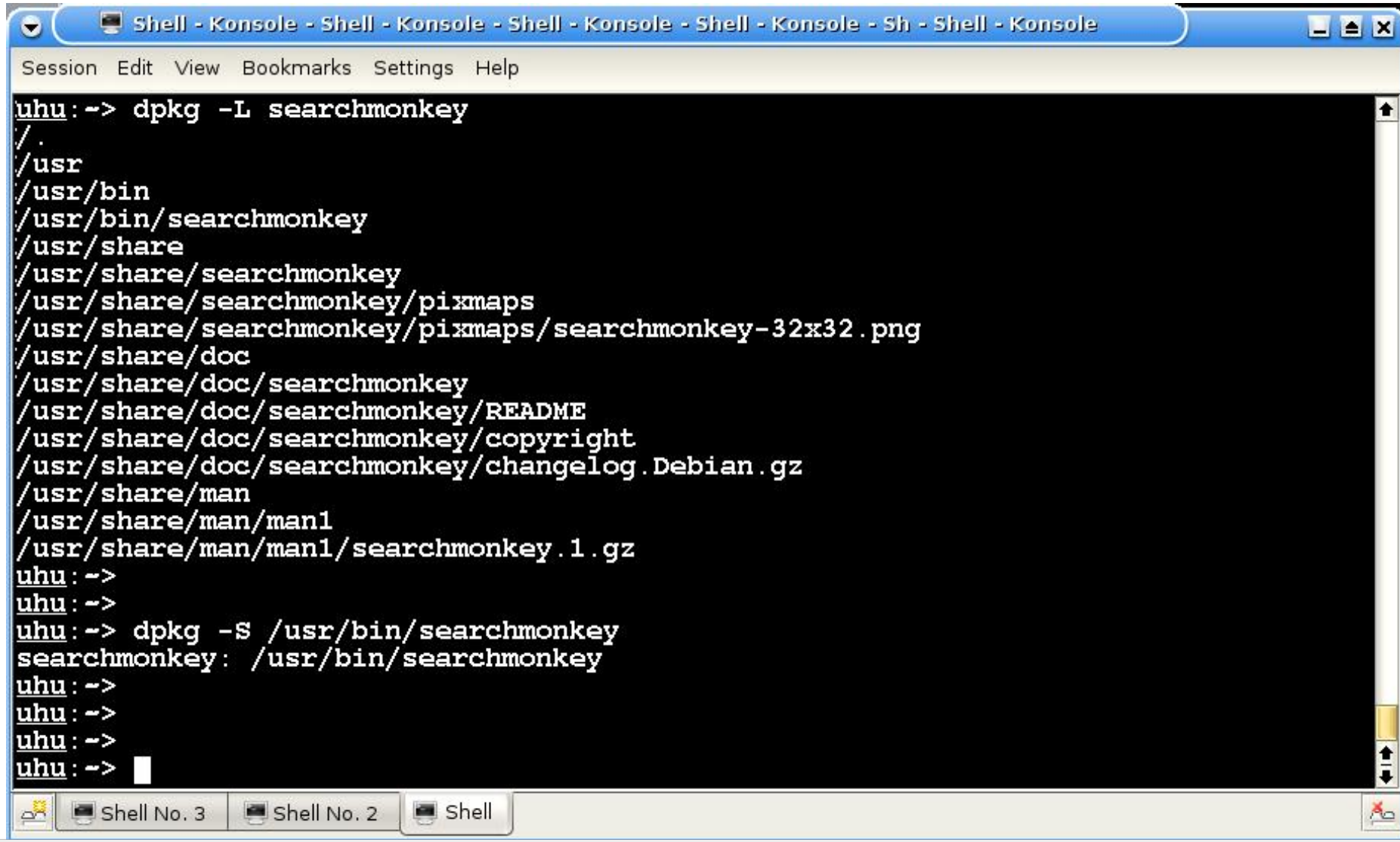
Working on a command line

Installing Debian packages - search for a package name



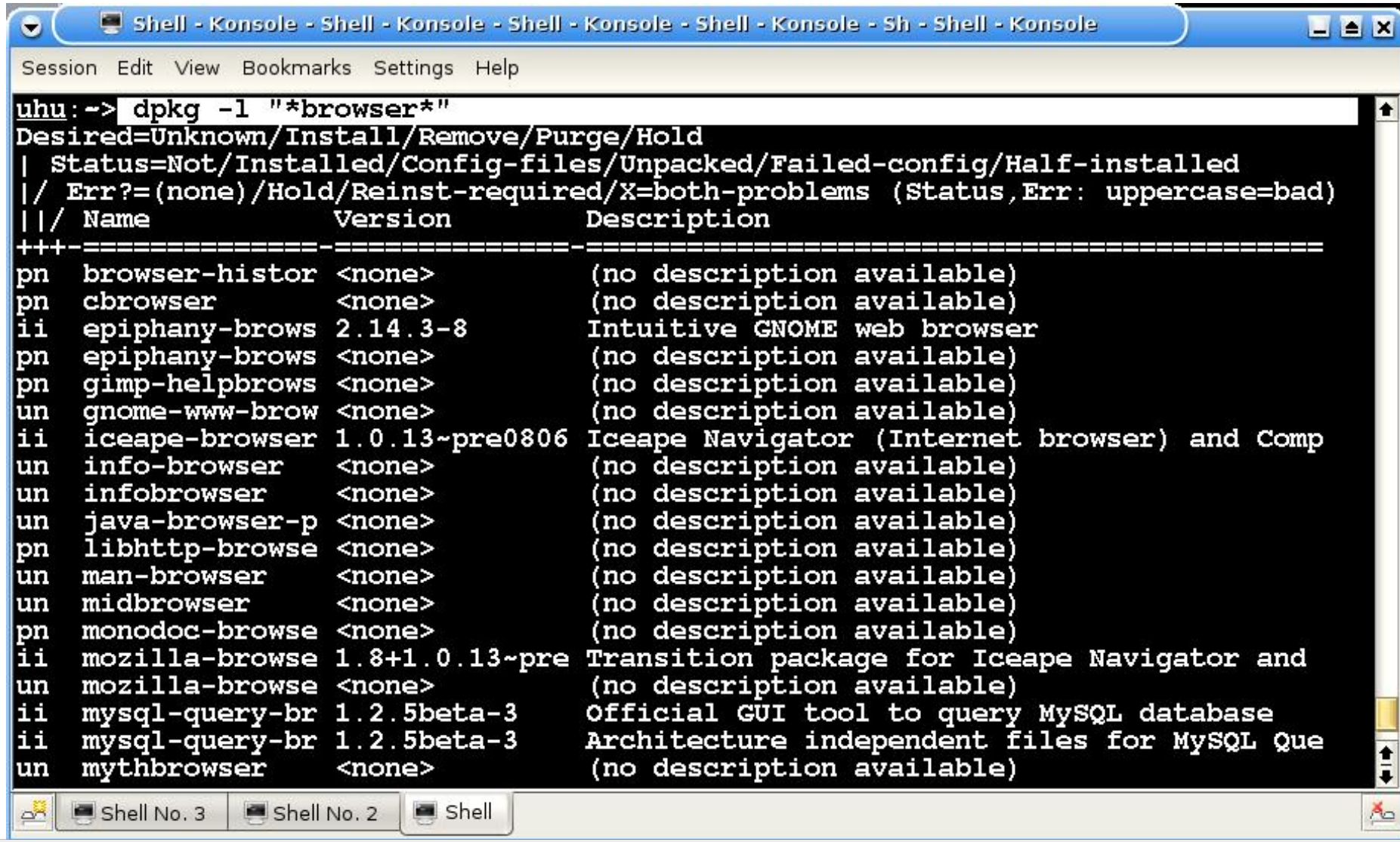
Working on a command line

Installing Debian packages - L list contents, -S search for name



Working on a command line

Installing Debian packages - list packages



Agenda

Working on a command line

Changing a user account by commands

chfn; chsh: change real user name and information; login shell

- chfn –f fullname –r roomno –w workph –h homeph –o other user
- chsh –help –shell SHELL [LOGIN]

YP Commands: yppasswd, ypchfn, ypchsh, ypcat, ypwhich, ypmatch

- yppasswd [-f] [-l] [-p] [user]
  - -p passwd; -l login shell; -f login data
  - ypchfn [user]; ypchsh [user] (is a link to yppasswd)
- hades 4> yppasswd -l ke

```
Changing NIS account information for ke on atlantis.risc.uni-linz.a
Please enter password:
Changing login shell for ke on atlantis.risc.uni-linz.ac.at.
To accept the default, simply press return. To use the
system's default shell, type the word "none".
Login shell [/bin/bash]: /bin/tcsh
The login shell has been changed on atlantis.risc.uni-linz.ac.at.
hades:5>
```

Working on a command line

yp-commands

ypcat

```
ypcat passwd |grep abaumgar
ypcat auto.home |grep abaumgar
ypcat auto.scratch |grep abaumgar
```

ypmatch

```
gorilla:sysadmin!1> ypmatch abaumgar passwd
abaumgar:x:10211:10020:Alexander Baumgartner:/home/abaumgar:/bin/tcsh

gorilla:sysadmin!4> ypmatch abaumgar auto.home
atlantis:/zlocal/sdb10:abaumgar

gorilla:sysadmin!5> ypmatch abaumgar auto.scratch
ibis:/zlocal/sda3:abaumgar
gorilla:sysadmin!6>
```

Account - Permissions

Using groups

```
atlantis:~> ls -ld /home/doench/  /home/khan
drwxr-x--- 2 doench doench 4096 Oct 29 17:35 /home/doench/
drwxr-x--- 2 khan khan 4096 Oct 29 17:35 /home/khan
atlantis:~>
atlantis:~>  ypcat passwd | grep doench
doench:Fusb8x9TDxDuc:13144:13144:Christian Doench:/home/doench:/bin/tcs
atlantis:~>
atlantis:~>  ypcat group | grep 13144
doench:*:13144:ke
atlantis:~>
atlantis:~> ypcat group | grep khan
khan:*:13134:ke
student:*:10030:ablinger,sinka,tec,rahkooy,khan,ghira,galea,vele,
korporal,korbelar,zaf,velkov,conran,sharkey,doench,wiesinge
atlantis:~>
```

Account - Permissions

Changing permissions by commands

chmod [ugoa]*([-+=]([rwxXst-]*|[ugo]))+

- a: all (u+g+o)
- chmod g-rwx,o-rwx /home/kerdei
- chmod 700 /home/kerdei/private (4: read; 2: write; 1: execute; 0: no right)

default setting for directory: drwxr-xr-x

```
sysadmin!16> mkdir junk
sysadmin!17> ls -ld junk
drwxr-xr-x 2 sysadmin sysadmin 4096 Oct 31 21:20 junk/
sysadmin!18> chmod g-r,o-r junk
sysadmin!19>
sysadmin!19> ls -ld junk
drwx--x--x 2 sysadmin sysadmin 4096 Oct 31 21:20 junk/
sysadmin!21>
```

Account - Permissions

Changing permissions by commands

```
sysadmin!20> touch junk/file1.txt
sysadmin!21>
sysadmin!21> ls -l junk/file1.txt
-rw-r--r-- 1 sysadmin sysadmin 0 Oct 31 21:20 junk/file1.txt
sysadmin!22>
hades:sysadmin!22> sudo su - ke
hades:1> cd /home/sysadmin
hades:2> ls -l junk/fi
junk/ unreadable
hades:2> ls -l junk
ls: cannot open directory junk: Permission denied
hades:3>
hades:3> ls -l junk/file1.txt
-rw-r--r-- 1 sysadmin sysadmin 0 Oct 31 21:20 junk/file1.txt
hades:4> ls -ld junk/
drwx--x--x 2 sysadmin sysadmin 4096 Oct 31 21:20 junk//
hades:5>
```

Account - Permissions

Changing active group

Special files and commands

- /etc/passwd
  - defines only one group, your main group of default group
  - as PhD students your have risc as the default group
- /etc/group - contains groups in the system
- groups - list of groups you belongs to
- Creating files, directories
  - the object's group will be your default group
- newgrp
  - newgroup groupname
  - changes your main group to the defined one
  - group cbwe2013 contains all PhD students at RISC
  - newgroup cbwe2013 changes your active group to that one
  - all files, directories you create will have this group

Agenda

Text editors
vi

vi modes: command mode, editing mode

- change to command mode: ESC from editing

command mode

- file: read: :e file; write: :w file reread: :e! file; insert: :r file
- file: exit: :q, ZZ; starting vi: vi filename;

change to edit mode

- i - insert text; a append text to line;

editing commands

- 5yy - mark 5 lines; p insert marked lines; r2w - replace two words;

advantage of vi for emergency case:

- you can always connect to Linux computer by ssh
- you can always use vi in the terminal window

Agenda

The grep/egrep command

grep [OPTIONS] PATTERN [FILE...]

```
grep searches the named input FILEs (or standard input if no files
are named, or if a single hyphen-minus (-) is given as file name)
for lines containing  a  match to the given PATTERN.
```

- By default, grep prints the matching lines.

Matcher Selection options

- E, –extended-regexp
  - Interpret PATTERN as an extended regular expression

The grep/egrep command

Options

Matching Control options

- -i, –ignore-case (Ignore case distinctions in both the PATTERN and the input files)
- -v, –invert-match (Invert the sense of matching, to select non-matching lines)
- -R, -r, –recursive (Read all files under each directory, recursively)
  - for big hierarchy structure and/or big files this option need lot of time to search all files !
- -f FILE, –file=FILE (Obtain patterns from FILE, one per line)
- -e PATTERN, –regexp=PATTERN (Use PATTERN as the pattern. This is useful to protect patterns beginning with hyphen-minus (-).

The grep/egrep command

Examples

Simple examples

- grep -i manager linux-all-packages.txt
- grep "File Manager" linux-all-packages.txt
- grep -i -e "file" -e "manager" linux-all-packages.txt
- grep -v " "
- grep -R -i Chau sysadmin/mails/archive.12

More examples

- check man page, internet sites, etc.

Regular Expressions

```
A regular expression is a pattern that describes a set of strings.
Regular expressions are constructed similar to arithmetic
expressions, by using various operators to combine smaller expressions.
```

Simplest expression

- Most characters, including all letters and digits, are regular expressions that match themselves.
- Any meta-character with special meaning may be quoted by preceding it with a backslash.

```
meta-characters: \{ \} \^ \$ \. The period matches any single chara
```

Character Classes and Bracket Expressions

- A bracket expression is a list of characters enclosed by [and]. It matches any single character in that list; if the first character of the list is the caret ^ then it matches any character not in the list.
  - example: [0123456789] matches any single digit.

Regular Expressions

Character Classes and Bracket Expressions

Bracket Expressions

- Range expression
  - it consists of two characters separated by a hyphen
  - it is within a bracket expression
  - It matches any single character between the two characters, inclusive, using the locales collating sequence and character set.
- example
  - in the default C locale, [a-d] is equivalent to [abcd]
  - Many locales sort characters in dictionary order: [abcd] might be equivalent to [aBbCcDd]
- using
  - du -s /[bcel-ort-v]*
  - ls -ld /[l-ot-v]*

Regular Expressions

Character Classes and Bracket Expressions

Predefined names

- certain named classes of characters are predefined

  `[:alnum:], [:alpha:], [:digit:], [:lower:], [:space:], [:upper:]`
  `[[:alnum:]] means [0-9A-Za-z]`

`Note that the brackets in these class names are part of the symbolic names, and must be included in addition to the brackets delimiting the bracket expression.`

Bracket Expressions

- Anchoring

  `The caret ^ and the dollar sign $ are meta-characters that match respectively the empty string at the beginning and end of a line.`

- The Backslash Character and Special Expressions

  `The symbols \< and \> respectively match the empty string at the beginning and end of a word.`

Regular Expressions

Character Classes and Bracket Expressions

Repetition

- Regular expression may be followed by one of several repetition operators:

```
?      The preceding item is optional and matched at most once.
*      The preceding item will be matched zero or more times.
+      The preceding item will be matched one or more times.
{n}    The preceding item is matched exactly n times.
```

Examples

- edit the tvlist.txt file in vi
- use regular expressions to get a readable version of the file

Regular Expressions

Bracket Expressions

Input lines

```
< value='9'>010 ZDF HD</option>
```

- commands with regular expressions in vi

```
:%s/option//g
:%s/>$//
:%s/\///
:%s/<$//

:%s/<\/option>$//

:%s/^< value=//
:%s/^'//
:%s/[0-9]*//
:%s/^'>//

:%s/^< value='[0-9]*'>//
```

Regular Expressions

Operators

Operators

- Concatenation

  Two regular expressions may be concatenated; the resulting regular expression matches any string formed by concatenating two substring that respectively match the concatenated expressions.

- Alternation

  Two regular expressions may be joined by the infix operator |; the resulting regular expression matches any string matching either alternate expression.

- Precedence

  Repetition takes precedence over concatenation, which in turn takes precedence over alternation. A whole expression may be enclosed in parentheses to override these precedence rules and form a subexpression.

Regular Expressions

Repetition, etc.

Examples

```
abc|def Finds either abc or def
abc+    Finds abc or abcc or abccc or ...
abc?    Finds ab or abc
abc*    Finds ab or abc or abcc or abccc or ...
abc$    Finds abc that occurs at the end of a line
^abc    Finds abc that occurs at the beginning of a line
[abcx-z]    Finds any single character of a, b, c, x, y, or z
[^abcx-z]    Finds any single character except a, b, c, x, y, or z
```

Regular Expressions

Examples

Examples in .procmailrc file

```
DEFAULT=/var/spool/mail/ke

:0 c
* ! ^X-Spam-Flag: YES
| /usr/bin/vacation -a K.Erdei ke

:0:
* ^From:(.*The System Administration|.*sysadmin|.*Super-user|.*root)
$DEFAULT
```

Agenda

The Shell - Main Features

The User Interface to the OS

Shell features

- an ASCII terminal window will be started
  - like in XP the DOS window (run - cmd)
- shell versions
  - sh, csh, bash, tcsh; others; see the man pages of the shells
  - at RISC default is the tcsh; echo $SHELL
  - we'll use/learn the bash, too
  - you can change your login shell
- miscellaneous parameters will be set per default
- environment variables: inherit values to sub-shells
  - list with printenv; set with setenv VARIABLE value
- local variables: scope only for the active shell
- lot of internal commands; invoking external (OS) commandos, too
- programing language: powerful shell scripting possible
- man sh: 4918 lines (80 pages)

The Shell - Redirection

Input/Output

Input/Output redirection

- standard input: console; standard output: display
  - minus (-) means standard input/output
- input output redirection;
  - ls -l > junk; (creates file junk and writes the output of ls -l in it)
  - cat jj >> junk; (appends the contents of the file jj to junk)
  - grep "txt" < junk | wc; (pipes the content of the file junk into the grep command and the output will be piped to the wc command (word count))
- pipeline connects output/input of two or more subsequent processes:
  - ps auxw | grep sendmail | wc; (counts how many sendmail processes are running)
  - more pipelines: who | sort | lpr; (prints a sorted list of the output of who
  - processes executed parallel

The Shell - Redirection

Input/Output redirection - an example

tar - store and extract files from an archive file known as a tarfile

- tar cf deepsky.tar deepsky/; (creates the single file: deepsky.tar, as archive file)
- tar cf - deepsky/ | (cd /scratch/ke/pictures/; tar xf -)
  - the contents of deepsky directory (tree structure) will be archived as a single file and will be piped to the standard output
  - the parentheses ( ) create a single command
  - first the current directory will be changed to the given one
  - second the standard input will be piped to the tar command and the archive will be extracted. In this process the same directory tree will be created as the original one.
- tar cf - deepsky/ | ssh bullfrog "(cd /scratch/ke/pics; tar xf -)"
  - the first process is identical with the above one
  - the second process is an ssh command, logging in a remote host.
  - the ssh command reads as input the standard input, which will be piped to the tar command

The Shell - Processes

Process management

Commands for the Process management - background/foreground processes

- commands for listing/killing processes:
  - ps auxw ; ps auxw | grep pattern
  - jobs
  - kill -TERM process-number
  - kill -9 process-number
  - kill % number; killall
- starting a command in foreground / background
  - start in background: & (command line available)
  - start in the forground: without & (command line is not available for work)
  - stop %job | number - stop the current job in foreground: $\hat{Z}$
  - bg %job | number
  - fg %job | number

The Shell - Processes

Process management

Limit/change/set some resources (tcsh)

- limit [-h] [resource [maximum-use]]
  - limits the given resource for the (next) started processes
  - resources: cputime, memoryuse, filesize, concurrency, maxproc
  - maximum-use: floating point or integer number followed by a scale factor
  - scale factor: for cpu use: second; for all others: k(byte), m(byte)
- nice [+number] [command]
  - run a program with modified scheduling priority

Limit/change/set some resources (OS)

- renice priority [[-p] pid ...] [[-u] user ...]
  - alter priority of running processes

Limit/change/set some resources (bash)

- ulimit [-HSTabcdefiIlmnpqrstuvx [limit]]

The Shell - Environment Variables

Builtin commands, loops

Alias in tcsh/bash

- tcsh: alias [name [wordlist]]
  - alias ar acroread
  - alias psh "ps auxw | more | grep ssh"
- bash: alias [name=[wordlist]]
  - alias psh="ps auxw | grep ssh"

Commands for defining loops (tcsh)

- foreach name (wordlist) ... end

```
foreach i (*)
echo $i
end
```

- Lot of other commands are available:
  - while, break, continue, repeat, count etc.

The Shell - Loops

Example loops

Example

- create a subdirectory in your home as pdf-files-2013-Oct
- copy all pdf files from your home to that directory

Solution tcsh

```
cd; pwd; mkdir pdf-files-2013-Oct
foreach i in (*pdf)
echo $i
cp -p $i pdf-files-2013-Oct
end
ls -1 pdf-files-2013-Oct | wc
```

Solution - trivial

- cd; pwd; mkdir pdf-files-2013-Oct; cp *pdf pdf-files-2013-Oct;

The Shell - Environment Variables

Looping in bash

Commands for defining loops (bash)

- for name [[in [word ...]] ;] do list ; done

```
for i in a b c  ;
do
echo $i;
done
```

Lot of other commands are available for cycles:

- for ((expr1 ; expr2 ; expr3)) ; do list ; done
- case word in [[(] pattern [— pattern] ...) list ;;] ... esac
- while list; do list; done
- until list; do list; done

The Shell - Builtin commands

if structures in tcsh

```
      if (expr) command
      if (expr) then
      ...
      else if (expr2) then
      ...
      else
      ...
      endif
while (expr) ...  end
switch (string)
      case str1:
          ...
          breaksw
      default:
          ...
          breaksw
endsw
```

The Shell - Builtin commands

switch structures in tcsh

```
switch ("$HOSTTYPE")
case *linux:
        alias lsl     ls -l
breaksw
case decstation:
        alias lsl     ls -lg
breaksw
case sun*:
        if ("`hostname`" == sun) then
                alias lsl     ls -lg
        else
                alias lsl     ls -l
        endif
breaksw
default:
        alias lsl     ls -l
breaksw
endsw
```

The Shell - Builtin commands

other commands in tcsh

Variables, environment variables and commands on it

- set - manages local variables
  - set (print all values)
  - set name ... (delete value)
  - set name=word ... (assign value)
  - unset pattern (delete variable)
- setenv - manages environment variables
  - setenv [name [value]]
  - unsetenv pattern

Shell config file: /home/username/.cshrc

- .cshrc will be executed by starting a new shell
- add command here to simplify or define work
- activate changed .cshrc file
  - source filename; source .cshrc

The Shell - configuration file

the .cshrc file - an example

Commands in the .cshrc file

```
set path=($path /zvol/timer/bin)

alias ls ls -F
alias ll ls -latr
alias psh "ps auxw|grep ssh"

alias crutch ssh -f -N -L 3389:crutch.risc.jku.at:3389 gorilla.risc.jku

alias bt ssh -f -N -L 1993:localhost:993 bullfinch.risc.jku.at
alias promt ssh -l kesysadm -f -N -L 2993:localhost:993 prometheus.risc

alias smtp ssh -f -N -L 3025:smtpauthhost.risc.jku.at:25 narwal.risc.jk
```

The Shell - Environment Variables

List of the environment parameters

Environment variables

```
SHELL=/bin/tcsh
HOST=uhu
USER=ke
GROUP=ke
HOSTTYPE=i486-linux
PATH=/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:
/zvol/timer/bin:/home/ke/bin:
DESKTOP-SESSION=kde
PWD=/home/ke
LANG=en-US.UTF-8
HOME=/home/ke
OSTYPE=linux
VENDOR=intel
LOGNAME=ke
MACHTYPE=i486
DISPLAY=:0
TERM=xterm
```

Working on a command line

Links to Wikipedia

Other shells

- Bash - Bourne Again Shell
  - man bash
  - .bashrc
- Learning how to use .bashrc
  - ypcat passwd —grep bash
  - ls -l /loginname/.bashrc
  - more /loginname/.bashrc

Links to Wikipedia

- http://en.wikipedia.org/wiki/List_of_Unix_programs
- http://www.think-lamp.com/2008/11/very-useful-linuxunix-commands/

End of Working on a Command Line

Thanks for your attention !