



# Computing at DESY Zeuthen

---

- an introduction -

Stephan Wiesand

DESY - DV -

July 27, 2006

# Content of this talk



- Part I

- computing environment
- policies
- resources
  - desktop PCs (linux)
  - login hosts & farms
  - storage, AFS basics
- getting started
  - basic shell usage
  - email, printing
  - application software

- Part II

- advanced shell usage
  - options, aliases
  - scripting
  - pipelines
  - I/O redirection
- more about AFS
- building software
  - compiling & linking
  - make
  - debugging



# Getting help

- central email address for questions & requests:  
[uco-zn@desy.de](mailto:uco-zn@desy.de)
- mail to this address
  - is read by all who can help
  - will create a ticket in our request tracker
    - your question can't get lost or be forgotten about
  - is usually answered very quickly
- do **not** mail questions to individuals
  - we are travelling or on leave occasionally
- note there's **no 24x7 service**



# Writing to uco-zn

- **bad** examples:
  - "I want to compile a programme and it doesn't work."
  - "My PC doesn't work properly."
  - "I see strange fonts."
- **good** example:
  - "I want to build a programme using the ROOT framework, version 5.08.00, using the default compiler. Builds fail with an error message from the linker about missing symbols. I'm working on lx64.ifh.de. I include my Makefile and the full output of the make command below..."



# Finding information

- **computing web pages:**  
`http://www-zeuthen.desy.de/computing/`  
`http://dvinfo.ifh.de`
- **this talk:**  
`http://www-zeuthen.desy.de/~wiesand/intro/`
- "Unix@desy.de" reference guide **booklet**
- **unix commands**
  - **man**
    - ("see frobnitz(4)" means "run man 4 frobnitz")
  - **info**
  - many commands have a **--help** switch

# Our computing environment



- major platforms:
  - **Windows** XP/2003
    - Desktops, a few servers (for Windows desktops)
    - Windows 2003 Terminal server
      - ICA (Citrix Metaframe)
        - see [http://www.ifh.de/computing/projects/win\\_desy\\_de/ICAinfo.html](http://www.ifh.de/computing/projects/win_desy_de/ICAinfo.html)
      - rdesktop(1) - try **winrdp**
  - **Solaris**
    - few login machines left, many backbone services
  - **Linux** (Scientific Linux 3/4)
    - desktops (yours is probably one of them)
    - farms, login hosts, services (i386 or amd64 platform)



# NB: Parallel Computing at Zeuthen

- **APE**
  - special purpose hardware
  - custom design & build
  - does one thing well: Lattice QCD
- **PC clusters**
  - more versatile
  - actually used for Lattice QCD as well
  - fast interconnect (Myrinet or Infiniband)
    - main difference w.r.t. "farm"



# Policies 1: Security

- DESY is an attractive target for hackers
  - and constantly under attack
- cracking a host is much easier from a DESY user account than from outside our network
- => please protect your account!
  - use a strong password
  - keep it to yourself
    - don't write it down
    - don't store it anywhere
    - don't share it with anyone



# NB: What's a **strong password** ?



- 7 to 8 characters long (avoid "\", "#", quotes, spaces)
- consists not only of lower case letters but
  - also characters from at least two of
    - digits
    - upper case characters
    - other printable characters
- is not vulnerable to **social engineering**
- **bad examples**: ih8\_pcs Isabe11a 2fast\_4U
  - many will be rejected automatically, some won't
- **good example**: g{XP52k

# Security policies continued



- don't install or run applications that accept or keep up network connections
  - except those provided by us
- don't run hacker tools, or try to hack hosts
  - contact [security@ifh.de](mailto:security@ifh.de) if you think you spotted a security problem
- don't change the permissions of your home directory
- don't connect notebooks to our network in place of a desktop, don't invent and configure IP addresses
  - [dynamic dhcp](#) is available on many wall sockets

# Policies 2: Acceptable use



- DESY computing resources are for research and education only
- no commercial activities!
  - mass mailings, web shops, ...
- no political campaigns!
  - again, neither through e-mail nor web nor ....
- don't consume CPU cycles, storage, bandwidth, ...
  - for anything but your work
    - even then, don't waste them unnecessarily
- no pirated materials! (movies, MP3s, software, ...)

# Resources: Your desktop PC



- purpose: local login, interactive work
  - mail, web, authoring, software development & tests
  - interactive data analysis
- login is also possible from other hosts
  - only to group PCs, not to guest PCs
  - not from outside DESY
  - some things work only when logged in locally
- local disk/CPU are **not highly reliable/available!**
- **home directory** is (it resides in **AFS**)



# PCs: SL3 vs. SL4

- this is a time of transition
  - Scientific Linux 3 is the established environment
  - Scientific Linux 4 is just emerging
- if in doubt, consult `/etc/redhat-release`
- SL4 desktop PCs: `{satyr<n> | n >= 60}`
- differences w.r.t. SL3 are many, among those:
  - removable media mount points
    - usually in `/media`; check `/etc/fstab`
  - C++ ABI (ROOT, CLHEP, GEANT4, ...)
- if in doubt, use some SL3 system to work there



# Resources: Storage

- Data storage is available in many flavours & qualities:
  - **AFS**
    - **secure** (not accessible without knowing the right password)
    - **redundant & highly available**
  - **Tape**
    - as secure as your account
  - **NFS**
    - **insecure** if exported to desktops
    - may or may not be redundant or highly available
  - **local disks** = scratch space, for convenience only
    - **insecure & volatile**



# More on storage

- availability of backup:
  - always **assume there is none**
  - except if explicitly stated otherwise
- your **home directory**
  - is backed up daily, has a snapshot taken every night
    - available in `~/OldFiles`
  - has a low quota (50 MB), can be raised on request
- **AFS/NFS group space**
  - is available from your group admin (ask backup status)
- **local disks** are **scratch space only!**



# What to store where

	<i>Home directory</i>	<i>AFS group space</i>	<i>NFS group space</i>	<i>Tape</i>	<i>local disks</i>
source code	yes	with backup	with backup	ok	no
compiled code	no	without backup	ok	no	ok
test data	no	ok	ok	no	ok
bulk data	no	without backup	without backup	ok	copy
shared access	no	yes	no	no	no
confidential	yes	yes	no	no	no
precious data	yes	with backup	with backup	ok	no
ripped DVDs	no	no	no	no	no

- data on local disks may vanish anytime
- it is not accessible from any other system





# Storage Locations

- **AFS**
  - homedir: `/afs/ihf.de/user/<initial>/<user>`
  - group space: `/afs/ihf.de/group/<group>`
- **NFS**
  - `/net/<group>/data<n>`
  - `/nfs/<group>/data<n>` (preferred)
  - `/net/<group>/disk<n>`, `net/<server>/data<n>`
- **local disk**
  - `/usr1/scratch` (make yourself a directory there)
  - guest PCs (no special group) have `/tmp` only
    - automatically cleaned after 2 weeks



# Resources: Login hosts

- pub
  - public linux login
  - will divert you to the least loaded one of pub1-6
  - also accessible from outside
  - right place for doing things that need much memory
    - if your desktop PC doesn't have enough
    - example: OpenOffice
  - not the right place for lengthy, CPU intensive jobs
    - use the farms for that
  - not the right place for storing data
  - not the right place for moving data



# Login hosts continued

- **lx64**
  - public **linux/amd64 test system**
  - use like pubs: test & evaluation only
- **dedicated login hosts**
  - most groups have some
  - typically Linux or Solaris
  - ask your group admin
- use **ssh <host>** to log in to <host> from your PC
  - forget rsh & friends
    - if you know them (if not: good)



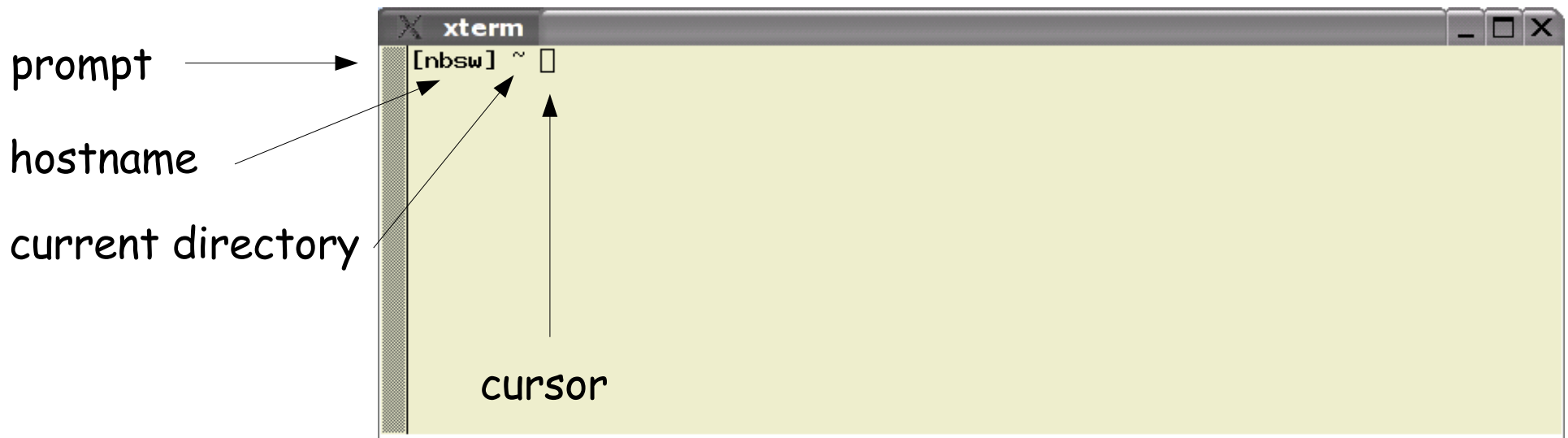
# Resources: Compute farm

- **PIII**: 80 CPUs, 800 MHz, 256-512 MB RAM/CPU
- **Opteron**: 246 cores, 2.2/2.4 GHz, 2-4 GB RAM/core
  - mostly 64-bit (amd64 aka x86\_64)
- common facility shared between all groups
  - **batch jobs**: simulation, data processing, ...
- **interactive access**: **qrsh**
  - heavy PAW/ROOT sessions, moving data, ...
- see [http://dvinfo.ifh.de/Batch\\_System\\_Usage](http://dvinfo.ifh.de/Batch_System_Usage) for details & instructions
- most **common mistake**: failure to request resources

# Getting started: Login, the shell



- windowing environments:
  - **Icewm, WindowMaker**: lean, low memory usage
  - **GNOME/KDE**: more gadgets, more point'n'click
- choose one on the login screen
- in either case, always keep open a **terminal window**:





# The shell

- What you see:
  - `xterm` (or `konsole` or `gnome-terminal`)
    - is a child process of the window manager
    - displays the window
  - the shell: `zsh` (`tcsh` is not recommended, `bash` not supported)
    - is a child process of `xterm`
    - prints the prompt (actual display is handled by `xterm`)
    - accepts and executes your commands
      - starts child processes
    - is your most important interface to the system



# Running commands

- in the **foreground**: type the command, hit return
- in the **background**: append **&** to command
- **jobs** will show current background commands
- **fg [%<n>]** brings job n back into foreground
- hitting **^Z** suspends a foreground command
- **bg** continues suspended command in the background

job number →

^Z →

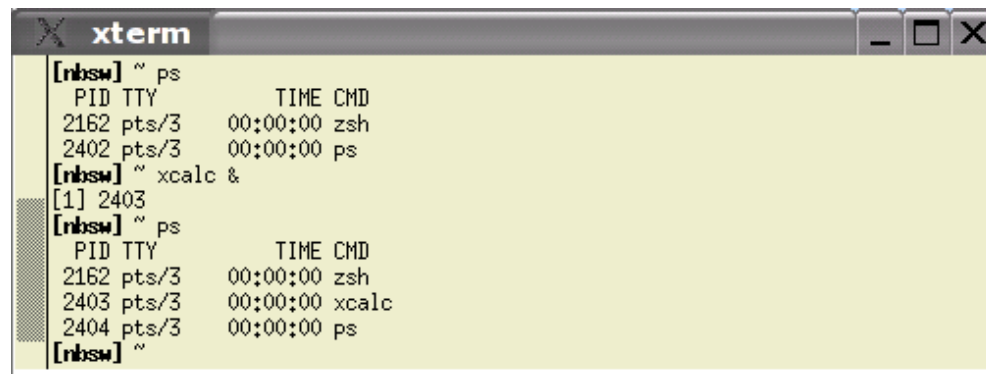
```
xterm
[nbsw] ~ xcalc &
[1] 2173 ←
[nbsw] ~ jobs
[1] + running  xcalc
[nbsw] ~ fg
[1] + running  xcalc

zsh: 2173 suspended  xcalc
[nbsw] ~ bg
[1] + continued  xcalc
[nbsw] ~
[1] + 2173 done    xcalc
[nbsw] ~ []
```

Process ID (PID)

# Processes

- **ps** shows processes (also try **top** and **qps**)
  - many options, try: **ps aux** (shows all processes)
- **kill** can send a **signal** to a process
  - **kill -<SIGNAL> <PID>**
  - useful signals include
    - **STOP** (suspend), **CONT** (continue)
    - **HUP** (hangup, kills softly), **TERM** (terminate), **KILL**

A terminal window titled 'xterm' showing the execution of 'ps' and 'xcalc &' commands. The output of 'ps' shows a list of processes with columns for PID, TTY, TIME, and CMD. The first 'ps' command shows processes with PIDs 2162 (zsh) and 2402 (ps). The second 'ps' command, after running 'xcalc &', shows an additional process with PID 2403 (xcalc) and a new PID 2404 (ps).

```
[nbsw] ~ ps
PID TTY      TIME CMD
2162 pts/3    00:00:00 zsh
2402 pts/3    00:00:00 ps
[nbsw] ~ xcalc &
[1] 2403
[nbsw] ~ ps
PID TTY      TIME CMD
2162 pts/3    00:00:00 zsh
2403 pts/3    00:00:00 xcalc
2404 pts/3    00:00:00 ps
[nbsw] ~
```





# The filesystem

- Unix filesystem is **hierarchic**, the **root directory** is /
- **directories** can contain files and directories
- a complete path is formed by **separating directory components by "/"** (not "\\"):
  - `/dir1/subdir2/subsubdir3/something`
    - `something` may be a file or a directory
- there's no small limit on the length of names
- most characters are allowed ( "/" isn't )
  - **avoid** those interpreted by the shell
    - `* [] {} ( ) \ | ; & ...`

# Special directories, navigating



- special directories:
  - `.` (a single dot): the current directory
  - `..` (two consecutive dots): the parent directory
  - `~` (only for the shell): your homedirectory
  - `~<user>` (only for the shell): someone else's
- command for moving in the filesystem:
  - `cd <path>` sets the shell's current directory
  - `cd ~` brings you home
  - `cd` does the same
  - `cd -` goes back to previous directory



# Copying and moving data

- `cp file [file ...] {file|directory}`
  - `copies` files to other files or into directories
  - `cp /some/path/fileA /other/path/fileB`
  - `cp /some/path/fileA /other/path`
    - same as `cp /some/path/fileA /other/path/fileA`
  - `cp fileA fileB ../fileC /some/directory`
    - copies three files
- `mv` works like `cp`, but `moves` files or directories
  - may not work across filesystem boundaries
- there is no `rename` command- use `mv`
  - `mv fileA fileB`

# Creating and deleting files/dirs



- `mkdir <path>` creates a directory
  - `mkdir /tmp/mydir`
  - `mkdir /tmp/mydir/mysubdir`
  - or: `mkdir -p /tmp/mydir/mysubdir`
- `rmdir <path>` deletes a directory
  - again, only works for the last component
- `rm <path>` deletes a file
- `rm -r <path>` recursively deletes directory trees
  - be careful!
- `touch <path>` creates an empty file



# Links - hard or symbolic

- `ln <file1> <file2>` creates additional directory entry
  - called a "hard link"
  - only works for files, not directories
    - and only within filesystems (AFS: within directories)
    - otherwise, usage is like for `cp`
- `ln -s <file1> <file2>` creates symbolic link
  - actually a different file pointing to the first one

```
xterm
[nbsw] /tmp/test echo foo >fileA
[nbsw] /tmp/test ln fileA fileB
[nbsw] /tmp/test ln -s fileB fileC
[nbsw] /tmp/test ls -l
total 8
-rw-r--r--    2 wiesand  sysprog    4 2003-07-13 10:20 fileA
-rw-r--r--    2 wiesand  sysprog    4 2003-07-13 10:20 fileB
lrwxrwxrwx    1 wiesand  sysprog    5 2003-07-13 10:20 fileC -> fileB
[nbsw] /tmp/test cat fileB
foo
[nbsw] /tmp/test cat fileC
foo
[nbsw] /tmp/test
```



# Examining files

- `cat <textfile>`
  - dumps content of text files
- `less <file>`
  - allows navigation (arrow keys, ...)
  - can handle many other formats besides text
    - most anything that can be converted to text
    - including directories, rpms, and many more
- `file <file>` shows the `type` of the file
  - educated guess only, type is not stored in filesystem
  - unix files are just a stream of bytes

# Listing files & permissions (mode)



- `ls <path>` lists files
  - `ls -l <path>` shows details ("long" listing)
    - `-rwxr-xr-x 1 root root 74384 2003-03-14 03:00 /bin/ls`
      - permissions for user owning file (read, write, exec)
      - permissions for group owning the file
      - permissions for others
      - number of hard links
      - size, date and time
    - `drwxr-xr-x 2 root root 4096 2003-05-17 09:03 /bin`
      - a directory
      - note `r-x` is needed for reading, not just `r--`

# Changing modes and ownership



- `chmod <modespec> <file>` changes permissions
  - `chmod +x <file>` makes file executable for anyone
  - `chmod u+x <file>` makes file executable
    - for user owning file only
  - `chmod go-r <file>` makes file unreadable
    - for group and others
  - `chmod g+w <file>` makes file group-writable
- `chown <new owner> <file>` changes ownership
  - you're probably not allowed to do that
- `chgrp <new group> <file>` changes file's group





# Permissions in AFS space

- permissions explained so far work in traditional UNIX file systems
  - local disks, NFS (/net/..., /nfs/...)
- in **AFS** (/afs/...), things are different:
  - **permissions are per-directory**, not per-file
  - many **traditional mode bits**
    - are either **ignored**, or
    - have a **different meaning**
  - instead, there are **ACLs** (access control lists)
    - listed and manipulated with the `fs` command
  - you'll hear much more about AFS in Part II



# AFS tokens

- to **access** anything in AFS space, it must be (by ACL)
  - either world-accessible
  - or host-accessible for the host you're working on
  - or you need an **AFS token giving you permission**
- the latter is the most common case
  - includes your **home directory**
- you get a **fresh token** by typing your password
  - when you **log in**
  - when you **unlock the screen**
  - when you **run the kinit command**



# AFS token expiration

- an **AFS token** is actually a wrapped **Kerberos ticket**
- you get a normal Kerberos ticket with the token
  - grants passwordless access to mail, other hosts, ...
- **AFS tokens and Kerberos tickets expire**
  - after **25 hours**
- afterwards, many things wont' work anymore
  - opening new windows,...
- => **Problems ? First thing at all, check your token!**
  - **tokens** shows a list (**klist** shows all krb tickets)



# Getting started: email

- **pine** is the recommended email client
  - ancient looks, but very convenient and reliable
- stay away from **kmail**
- **thunderbird** can be used as well
  - but doesn't know about your kerberos ticket
    - and may require manual configuration
- mail server provides imap4/ssl - don't use **pop**
- save mails in home directory, **delete from inbox**
- **forwarding**: connect to
  - <http://registry.desy.de/registry>



# Getting started: printing

- to set your **default printer**, edit `~/ .zprofile`
  - it's prepared: `PRINTER=ps_lo2`
  - sets an environment variable (see Part II)
- printing **commands**:
  - `lpr [-P <printer>] <file>`
  - `lp [-d <printer>] <file>`
- a single printer may have multiple (queue) names
  - `hpcolor1/hpcolor1t` for **paper/transparencies**
- print in **color only if necessary**
  - often much more **expensive** than black & white

# Application software: editors



- **xemacs** is recommended
  - syntax highlighting for many programming languages
  - many other powerful features
- **emacs** (GNU emacs) is available as well
  - just as good and powerful
- **vi** or **gvim**
  - much leaner than emacs, just as powerful
  - but a matter of taste
- **nedit**
  - Windows addicts tend to like this one



# Other application software

- some software can only be used after an appropriate `ini` command
- a plain `ini` shows a list - some examples:
  - `ini pgi32`
    - modifies environment for using PGI compilers
    - `cc`, `f90` etc.
  - `ini ROOT`
    - allows using the `root` software (from CERN)
  - `ini ROOT_64`
    - use 64bit version of ROOT (on amd64 systems)
  - `ini -d ROOT_64` reverts the last change



# Math Software

- Maple
  - unlimited number of licenses
- Mathematica
  - limited number of licenses
    - don't waste
    - prefer Maple if you can
- Matlab
  - very limited number of licenses
    - each as expensive as a medium size car
  - don't expect one to be available, except from your group



# Storage media on desktop PCs



- CD/DVD-ROM
  - `mount /mnt/cdrom` makes content available there
  - run `umount /mnt/cdrom` before removing the media
- legacy floppy (may no longer work on all PCs)
  - `[u]mount /mnt/floppy`
- USB memory sticks
  - `[u]mount /mnt/hotplug` (1<sup>st</sup> parttion on device)
- never ever forget the `umount` command
  - or you will damage the `filesystem` on any r/w device
- a public CD writer is available in 2L01

# Remote commands & copying



- `ssh <host> <command>`
  - executes a command on <host>
  - works with many, but not all commands
  - default command is a login shell
- `scp [host1:]file1 [host2:]file2`
  - copies files between hosts (one must be local host)
  - mostly works like `cp`:
    - `scp pub3:/tmp/myfile ~`
    - `scp pub:/not/available/on/desktops/myFile /tmp`
  - wildcards (see Part II) must be quoted

# That's it for today



- Questions ?
- See you for Part II
- Have a pleasant and successful stay here at DESY Zeuthen!