Computing at DESY Zeuthen

An Introduction

Stephan Wiesand
Summer Students Lecture
Zeuthen, 2015-07-23

HELMHOLTZ GEMEINSCHAFT

DESY

Content

- 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
  - using the batch farm
  - building software
    - compiling & linking
    - make, debugging

Getting Help

- central email address for questions & requests: 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 5.14.00, 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 wiki/web pages:
  `http://dvinfo.ifh.de`
  `http://dv-zeuthen.desy.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 7/2008
    - Desktops, a few servers (for Windows desktops)
    - Windows 2008 Terminal server
      - `rdesktop(1) - try winrdp`
  - Linux (Scientific Linux 6/5)
    - all 64-bit
    - desktop PCs
    - login hosts, workgroup servers, farm, cluster, services
  - Solaris
    - very few machines left for special purposes

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 store it anywhere
    - don't share it with anyone
      - your account is for yourself only – do not share it
  - note your initial password must be changed within 5 days

NB: What's a Strong Password ?

- at least 8 characters long
- consists not only of lower case letters but
  - also characters from at least two of
    - digits
    - upper case characters
    - other printable characters
      - avoid "\", "#", quotes, spaces
- 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
- **Do not use your DESY password for anything else!**
  - **use separate passwords for google, facebook, twitter, ...**

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 if you think you spotted a security problem
- don't change the permissions (ACLs) of your home directory
- don't connect notebooks to our network in place of a desktop
  - it won't work, and will cause trouble
- don't invent and configure IP addresses
  - dynamic dhcp is available on WLAN & many wall sockets
  - your notebook can be registered for the guest network
  - eduroam is available too

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
  - possibly: software development & tests
  - possibly: interactive data analysis
- login over the network is not possible
  - if you (think you) need this, you're doing something wrong
- local disk/CPU are not highly reliable/available!
  - and they do break occasionally
- home directory is (it resides in AFS)

Linux: SL6 vs. SL5

- there are two major Linux releases in use:
  - Scientific Linux 6 is the default platform, 99% of all systems run it
  - Scientific Linux 5 is still available and supported, but frozen
    - a few WGSs, very few desktops
  - if in doubt, consult `lsb_release -r`
- there are some differences, among those:
  - software versions
    - in particular: default GCC compilers
      - C++ ABI (ROOT, CLHEP, GEANT4, ...) "mostly" compatible
    - should work: build on SL5, run on SL6
    - may not work: build on SL6, run on SL5
  - policies
    - desktop access, local firewall
- see `http://dvinfo.ifh.de/Linux_at_DESY_Zeuthen`

Resources: Storage

- Data storage is available in many flavours & qualities:
  - AFS
    - secure (not accessible without knowing the right password)
    - redundant & highly available
  - Lustre
    - fast parallel storage for use in batch jobs, analysis
  - Tape
  - NFS
    - insecure if exported to desktops; may or may not be redundant or highly available
  - local disks = scratch space, for convenience only
    - insecure & volatile
    - not exported to other systems

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 by default, can be raised on request (within limits)
- AFS/Lustre/dCache/NFS group space
  - is available from your group admin (ask backup status)
- local disks are scratch space only!
  - SL6 desktops and WGS only have /tmp
  - SL5 desktops may still have /usr1/scratch – but prefer group space

Storage Locations

- AFS
  - homedir: **`/afs/ifh.de/user/<initial>/<user>`**
  - group space: **`/afs/ifh.de/group/<group>`**
- Lustre (specific groups only)
  - **`/lustre/fs*/<group>`**
  - not available on PCs
- local disk
  - **`/tmp`** (make yourself a directory there)
    - automatically cleaned after 10 days

DESY

What to Store Where

	Home directory	AFS group space	Lustre/NFS group space	Tape	Local disk
source code	yes	with backup	with backup	ok	no
compiled code	no	yes	ok	no	ok
test data	no	yes	yes	no	ok
bulk data	no	without backup	without backup	ok	copy
shared access	no	yes	no	no	no
confidential data	no	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
- small files cause overheads
  - especially on Lustre, Tape

Resources: Login hosts

- pub(.ifh.de)
  - public linux login
  - will divert you to the least loaded one of pub1-6
  - also accessible from outside
  - 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
    - use transfer.ifh.de for that
  - these systems are old, and slow by today's standards
  - they should mainly be used to connect to a WGS
- warp(.ifh.de)
  - will transparently connect you to a workgroup server

Login Hosts continued

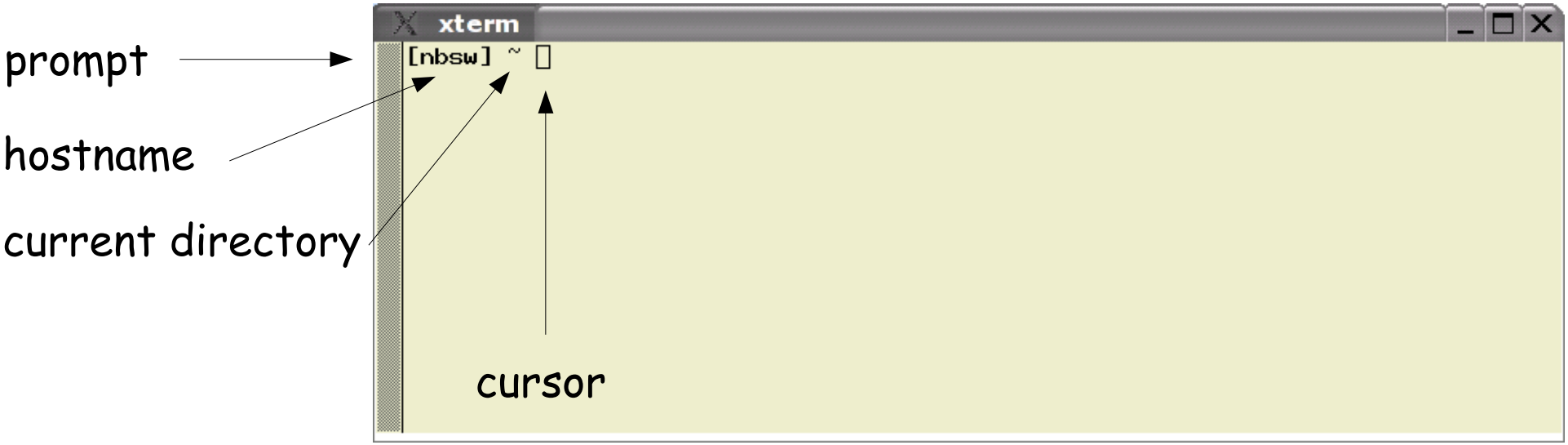
- lx64
  - Public SL5 linux test system
  - use like pubs: test & evaluation only
- sl6
  - a VM for testing SL6, same rules as for pub/lx64
- dedicated workgroup servers
  - most groups have some
  - use these instead of pub/sl6/lx64 if available to you
  - typically Linux, sometimes Solaris
  - ask your group admin
- use `ssh <host>` to log in to <host> from your desktop

Resources: Compute Farm & Cluster

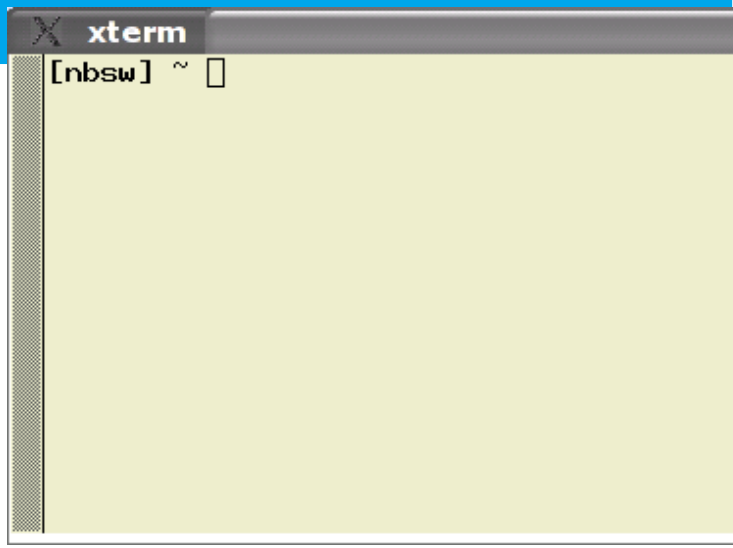
- all nodes run 64-bit SL6
- significant resources:
  - Farm: ~ 1700 cores, all Intel Xeon
    - 2.83 GHz Harpertown or faster, 4 GB available per core
  - Infiniband cluster: 1024 Cores
    - 2.8 GHz Nehalem or faster, 3 GB/core
    - restricted use
- common facility shared between all groups
  - batch jobs: simulation, data processing, ...
- interactive access: `qrsh`
  - heavy PAW/ROOT sessions, moving data, ...
- see `http://dvinfo.ifh.de/Batch_System_Usage`
- most common mistake: failure to request resources

Getting started: Login, the Shell

- windowing environments:
  - GNOME - recommended, except on very old PCs
  - Icewm - lean, low memory usage
  - KDE available as well (SL5)
- choose one on the login screen
- in either case, always keep open a terminal window:



The Shell



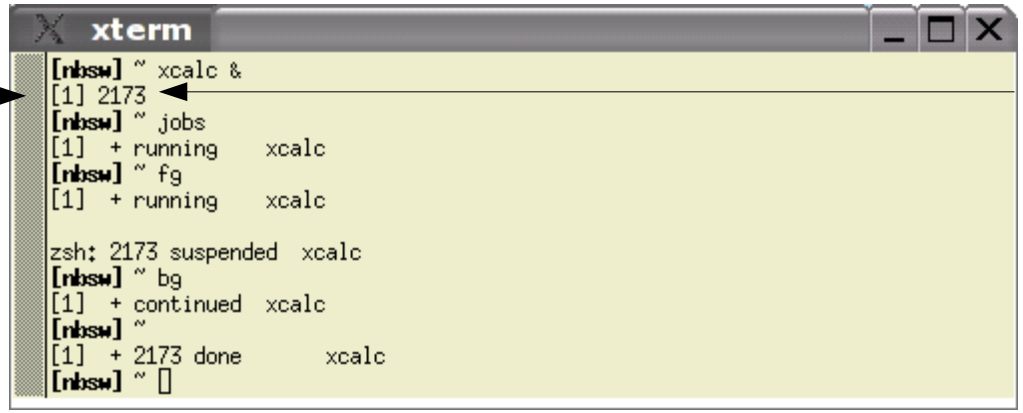
- What you see:
  - xterm (or konsole or gnome-terminal)
    - is a child process of the window manager
    - displays the window
  - the shell:
    - is a child process of the terminal application
    - prints the prompt (actual display is handled by terminal)
    - accepts and executes your commands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

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`** (Ctrl-Z) suspends a foreground command
- `bg` continues suspended command in the background

job number

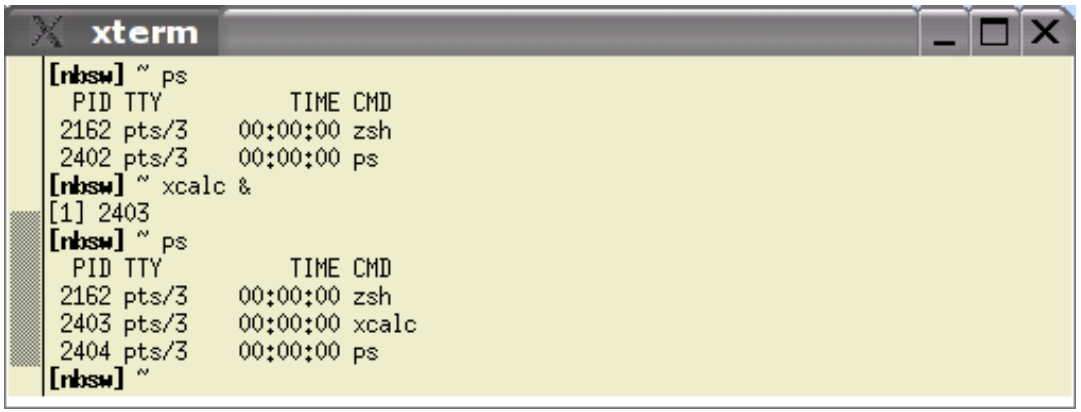
^Z



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



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 in names
  - "/" 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 work directory
    - path may be absolute, or relative to the current working 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 / Directories

`> 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

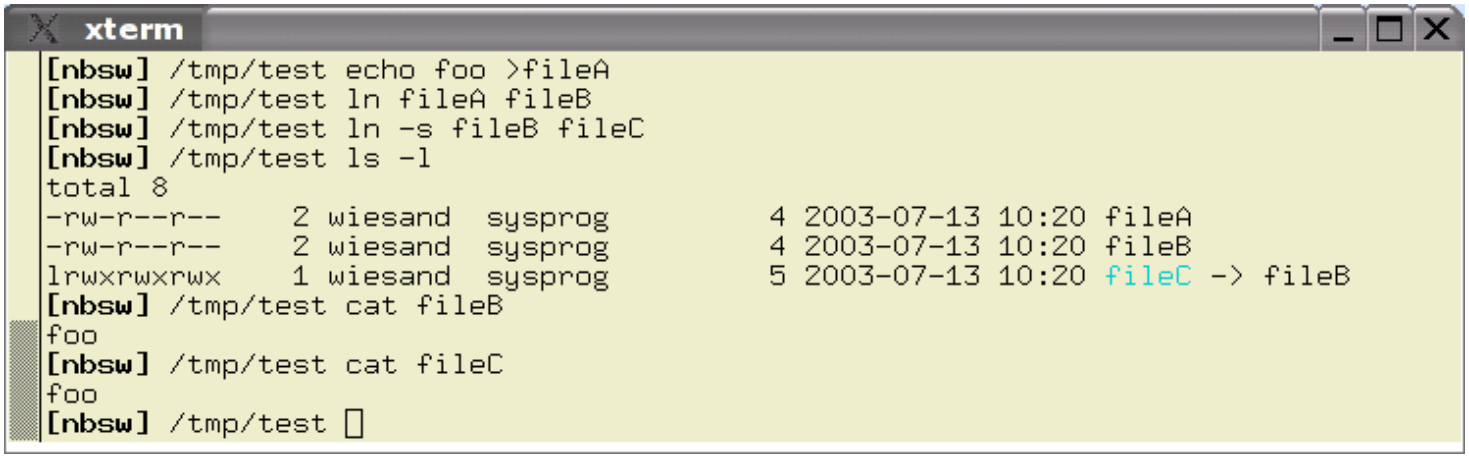
`> rm -r <path>` recursively deletes directory trees

- be careful!

`> touch <path>` creates an empty file

Links - Hard or Symbolic

- `ln` <file1> <file2> creates an 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 a symbolic link
  - actually a different file pointing to the first one



Examining Files

`> cat <textfile>`

- dumps content of text files on the terminal

`> 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 117024 Jun 22 2012 /bin/ls`
    - permissions for user owning the file (**r**ead, **w**rite, e**x**ec)
    - permissions for group owning the file
    - permissions for others
    - number of hard links
    - size, date and time of last modification
  - `drwxr-xr-x 2 root root 4096 Jun 23 05:17 /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 e**x**ecutable
    - for anyone
  - `chmod u+x <file>` makes file e**x**ecutable
    - for **u**ser owning file only
  - `chmod go-r <file>` makes file un**r**eadable
    - for **g**roup and **o**thers
  - `chmod g+w <file>` makes file **g**roup-**w**ritable
- `chgrp <new group> <file>` changes file's group
  - must be member of old & new groups, not allowed in AFS
- `chown <new owner> <file>` changes ownership
  - you're probably not allowed to do that

Permissions in AFS Space

- permissions explained so far work in traditional UNIX file systems
  - local disks, NFS, Lustre
- in AFS (/afs/...), things are different:
  - permissions are per-directory, not per-file
  - some of the 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
- tokens have a limited lifetime

AFS Token Expiration

- an AFS token is actually a wrapped Kerberos ticket
- you get a normal Kerberos ticket together 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 kerberos tickets)
- lock your screen when you leave it (or log out)
  - unlock will give you a fresh ticket/token on the local host
    - but not on other hosts you're logged into by ssh

Getting Started: e-mail

- `alpine` is one recommended e-mail client
  - ancient looks, but very convenient and reliable
  - should be preconfigured
- `thunderbird` can be used as well and is a common choice
  - but doesn't know about your kerberos ticket
    - and may require manual configuration
- `evolution` is another choice (most natural in GNOME)
- mail server provides imap4/ssl - don't use pop
- instructions:
  - https://dvinfo.ifh.de/MailReaderConfiguration
- forwarding: please avoid!

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>`
- the printing is CUPS
  - see `http://dvinfo.ifh.de/Printing_with_Cups`
- print in color only if necessary
  - much more expensive than black & white
    - even on color printers
  - preference: b&w on b&w printer, b&w on color printer, color

Application Software: Editors

- `emacs` is recommended
  - syntax highlighting for many programming languages
    - + other support (indentation,...)
  - many other powerful features
- `vim` or `gvim`
  - much leaner than emacs, just as powerful
  - but a matter of taste, hard for beginners
- `gedit`
  - maybe the best choice for beginners
- `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 pgi`
    - modifies environment for using PGI compilers
    - cc, f90 etc.
- on SL6, ini is being replaced with “environment modules”
  - `module avail`
    - shows a list
  - `module load intel/2013`
    - modifies environment for using Intel compilers
    - icc, ifort
- also available: xmaple13, mathematica9.0, ...

Math Software

- Maple
  - unlimited number of licenses
- Mathematica
  - very limited number of licenses
    - don't waste
    - prefer Maple if you can
- Matlab
  - very limited number of licenses
    - don't expect one to be available for you
    - prefer Maple or Mathematica if you can
- Please free licences as soon as possible
  - don't keep idle sessions running
- never consume more than a single license

Language Support, I18N

- the LANG environment variable controls application behaviour
  - sorting order, character display, ...
  - default on SL5 is **`LANG=C`**
  - default on SL6 is **`LANG=en_US.UTF-8`**
  - users can set **`LANG`** for themselves
    - in **`~/.i18n`** (->shell) and/or **`~/.dmrc`** (->window manager)
- typing special characters on US keyboards
  - with R-Alt as the "Compose Character" key
    - [R-Alt],["],[a] yields ä, also works for ç ø æ Å ñ ô é è €...
    - "european" languages, UTF-8 independent (ISO-8859-15)
  - using Input Methods, UTF-8 only
    - こんにちは , Дубна Ресторан (typed: Dubna Restoran)
- only English supported for UI

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

- See you tomorrow for Part II

- Have a pleasant and successful stay here at DESY Zeuthen!

- Questions ?