9a - Die Shell: Ihr persönlicher Kellner

9b - Das besondere an der Bash-Shell

- **a** ist nicht gleich **A**!!! z.B. XF86Config != xf86config
- Befehlspeicherung <Pfeil> nach oben
- ein speziefischer Befehl zurückholen:**<Strg>r** und dann **<Tab>** oder **<Return>**
- Cursor Positionierung:**<Strg>a** oder **<Pos1>**, **<Strg>e** oder **<Ende>**

Shells, Shell commands

Consoles under linux

- Standard Virtual Consoles (1-6)
- X-windows console processes (`xterm` & `bash` combination)

Good Terminal programs to use under X-Windows

Terminal	Installed?	SuSE Package	Serie
xterm	Yes	xf86	x
rxvt	no	rxvt	xap
wterm	no	wterm	xap
eterm	no	eterm	xap
aterm	no	aterm	xap
konsole	yes	kdebase	k2de

Use the arrow keys, and mouse middle key

Use of Page-Up Page-Down keys to scroll virtual terminal

Shell built-in commands vs. external commands(programs)

`man echo` Shows the man page of all internal bash functions

Shell description, use and types (bash,zsh,tcsh**)**

External programs running structure

- `Command [-`*`options`*`] [`*`parameters list`*`......]`
- Special manipulations of commands:

```
        command1 $(command2)
or      command1 `command2`
```

Use the result of `command2` as parameter of `command1`

Who can use which command

- su command to change user
- Use of PATH
- help on commands (**`man, info, apropos, which, whatis, whereis`**)

Changing the PATH for each user

- **~/.profile** script file includes the filling process of individual users PATH
- **~/.bashrc** Some more shaell parameters may be here

Mics shell commands

`<ESC>` *`key`*	Same as ALT-*key*. Useful for shells running under X-Terminals
`Ctrl-L`	Clears the screen (same as `clear` command)
`reset`	Resets a badly mangled terminal fonts and display

Line Editing commands (E-Macs commands- Readline Library)

`Ctrl-D`	Delete one CHAR to the right (Same as the DEL key)
`Ctrl-K`	Delete till end of line
`Ctrl-Y`	Paste text previously deleted
`<ESC> D`	Delete one WORD to right

Sequence of priority when a command is issued to bash:

1 - Look in **alias** list
2 - Look in **inbuilt functions** of bash
3- Look in user defined functions
4 - Look in directories contained in the **PATH** variable

echo Command:

echo -e	"....**\a**...."	Alert (bell) --Note: only in Virtual Terminal
echo -e	"....**\b**...."	Backspace
echo -e	"....**\c**...."	Suppress training new line
echo -e	"....**\f**...."	Form Feed
echo -e	"....**\n**...."	New Line
echo -e	"....**\r**...."	Carriage Return
echo -e	"....**\t**...."	Horizontal Tab
echo -e	"....**\v**...."	Vertical Tab
echo -e	"....\\...."	Litteral Baskslash \
echo -n	No Carriage Return at end of line	

Shell jobs Administration

Ctrl-Z To suspend (stop) a job and send it in the background

`bg` *`<JobNr.>`* To resume a suspended job that has been stopped with *Ctrl-Z*. It will behave as sif it got started with the &

`fg` *`<JobNr.>`* To resume a background job to the foreground

`jobs` *`<JobNr.>`* Display jobs own by present bash:

-l	List the process ID and Job Number
-n	Status since last job change
-r	List of running jobs only
-s	List stopped jobs only

`disown` Disown the activated(+) background job
(the child will no more receive a SIGTERM when the parent dies)

-a	Disown all the background jobs
-r	Disown only the running jobs
-h	Disown active job (+)from shell only when shell is closed: Disowned job will not die when shell dies.

Ubung:

Start `xterm`
in `xterm` start another `xterm` (**`xterm &`**)
close first `xterm`.....the second is **not** closed.
Start `xterm`
in `xterm` start `wterm` (**`wterm &`**)
Close `xterm`.....the `wterm` is also closed (owned by `xterm`)

Start xterm
in `xterm` start `wterm` (**`wterm &`**)
in `xterm` > **`jobs`** ....shows the background job
in `xterm` > **`disown`**... the last active job is disowned
Close `xterm`.....the `wterm` is NOT closed.

Bash Command History manipulations:

`history [`*`n`*`]` Displays the whole command histrory with history command numbers. Or the last ***n*** lines of the history
`HISTSIZE=x` Sets the maximum number of commands in history
`Command` `!$` Calls the command with the parameters of the last command
`!!` Recalls the last command from history
`!`*`n`* Recalls the command number '*n*' from history
`!-n` Recalls the last command minus 'n' commands from history
`!`*`String`* Recalls the most recent command <u>starting with</u> the *`String`*, from history
`!?`*`String`* Recalls the most recent command <u>containing</u> the *`String`*, from history.
`^` *`String1^String2`* Recalls the last command but replaces the *`String1`* by the *`String2`* (String1 must be a full word)

Recalling commands using the Keyboard

`Ctrl-P` Goes <u>up</u> in command history (Same as UP Arrow key)
`Ctrl-N` Goes <u>down</u> in command history (Same as DOWN Arrow key)
`Ctrl-B` Editing: Goes <u>left</u> in command (Same as LEFT Arrow key)
`Ctrl-F` Editing: Goes <u>right</u> in command (Same as RIGHT Arrow key)
`Ctrl-A` Editing: Goes at <u>start of line</u> in command (Same as HOME key)
`Ctrl-E` Editing: Goes at <u>end of line</u> in command (Same as END key)
`Ctrl-S` Search <u>forward</u> for command containing a specific text
`Ctrl-R` Search <u>backwards</u> for command containing a specific text
`<ESC> <` Go to Begining of command history
`<ESC> >` Go to End of command history

The find command (fc) program

`fc` Program to recall specific commands of Command history.
`fc -l` Displays the last 16 previous commands of the command history.
`fc` *`CmdNr`* Loads the command pointed by *`CommandNr`* in standard editor and executes the edited command.
`fc -s` *`CmdNr`* Recalls the command pointed by *`CommandNr`* the and executes it. (doesn't allow to edit it before running it)
`fc -l` *`command`* List the first found *command* in history and all the other ones after.
`fc -lr` *`command`* List the last found *command* in history and all the other ones after in reverse order.