

# UNIX summary of the most important commands

## 1. The Session

### 1.1 login

Type in username and password to prompt.

### 1.2 logout

type in “**logout**” or CTRL-d.  
(CTRL-d means: pressing Control and d keys simultaneously)

### 1.3 syntax of Unix-commands

command -options arg1 ... argn <RETURN>  
(also see man intro)

## 2. Users

who is logged in?

**who**

find out your own username:

**who am i**

**whoami**

**id**

find out username

**finger <username>**

## 3. Help

show help for commands  
window-based

**man <command>**

**xman**

other possibilities

<http://www.google.de>

## 4. Account

in which groups am i?  
how much disk space do i  
use, and how are my limits?  
disk usage for current directory  
... for directory <directory>  
show free disk space

**groups**

**quota -v <username>**

**du**

**du <directory>**

**df -H**

## 5. Paths

### 5.1 path names

simple A name for a file or directory, for getting access a file/directory in the current directory.  
absolute A list of directories, which begin with the root directory ‘/’, up to the the desired file/directory. Partial paths are separated with the help of ‘/’.  
relative A list of directories, which begin in the current directory. Partial paths are separated with the help of ‘/’.

### 5.2 changing between directories

print working directory

**pwd**

change to home directory

**cd**

change to any directory

**cd <dirname>**

### 5.3 abbreviations

home-directory

~

home directory of <username>

~<username>

present directory

.

predecessor of curent directory

..

## 6. Listing Files

in current (i.e. working) directory

**ls**

in other directory

**ls <dirname>**

list all information for files

**ls -al**

list all, sorted by time

**ls -alt**

show directories only

**ls -d**

## 7. inspecting files

show the complete file

**cat <filename>**

inspect file page by page

**less <filename>**

**pg <filename>**

show the first 20 lines

**head -20 <filename>**

show the last 20 lines

**tail -20 <filename>**

## 8. Security

### 8.1 passwords

Good passwords contain upper and lower cases, numbers, and special characters. They should contain at least 8 characters.

Bad passwords are names, words from dictionaries, phone numbers etc.

### 8.2 access rights

File	Directory
-rwx rwx rwx	d rwx rwx rwx
$\underbrace{\hspace{1em}} \underbrace{\hspace{1em}} \underbrace{\hspace{1em}}$	$\underbrace{\hspace{1em}} \underbrace{\hspace{1em}} \underbrace{\hspace{1em}}$
user group others	user group others

with r = read, w = write, x = execute

access rights are changed with:

**chmod** w?y <filename>  
w = u, g, o, a for user/group/other/all  
? = +, -, for adding/removing rights  
y = r, w, x for read/write/execute

alternative:

**chmod** abc <filename>

abc is specified as follows:

	a: user	b: group	c: other
read	4	4	4
write	2	2	2
execute	1	1	1

for desired rights, the columns are summed up.

example: **chmod** 760 <filename>

assigns the rights rwx rw- --- to the file <filename>

## 9. Creating Files

e.g. with the help of editor, file-redirection commands, or touch

**echo** 'Hello World!' > <newfile> creates a new file with content "Hello World!"  
**touch** <newfile> creates a new empty file

## 10. Creating Directories

**mkdir** <directory> creates a new directory with name <directory>

## 11. Copying Files

**cp** <oldfile> <newfile> copies <oldfile> to <newfile>  
**cp** <oldfile> <dirname> copies <oldfile> to directory <dirname> with the same filename <oldfile>

## 12. Renaming Files

**mv** <oldfile> <newfile> renames <oldfile>  
**mv** <oldfile> <dirname> moves the file <oldfile> to the directory <dirname>  
**mv** <olddir> <newdir> renames the directory <olddir>  
**mv** <olddir> <dirname> moves the file <olddir> into the directory <dirname>

## 13. Removing Files/Directories

**rm** <filename> deletes the file <filename>  
**rm** -i <filename> asks the user again, before <filename> is deleted (-i = interactive)  
**rm** -r <dirname> deletes a directory and all its content  
**rm** -ri <dirname> = rm with options -i and -r  
**rm** -f <filename> deletes the file <filename>, no interaction, opposite of -i  
**rmdir** <dirname> deletes empty directory

## 14. Printing Files

**lpr** -P<printrname> <filename> prints the postscript file <filename> on the printer <printrname>

**lpq** -P<pname> shows the printer-queue of printer <pname>

**lprm** -P<pname> <jobnr> deletes the print job with number <jobnr> from the printer queue of printer <pname>

**a2ps** <fname> is a program which converts a raw textfile into postscript format.  
→ not standard, may be not available

## 15. History

**history** shows the past commands

## 16. Redirection of In- and Output

command > <fname> redirects the output of the command to the file <fname>

command >& <fname> redirects the output of the command, as well as all error-output to the file <fname>

command >> <fname> appends the output of the command to the file <fname>

command < <fname> reads input from file <fname>

command1 | command2 (Pipe) Output of command1 is used as input for command2

command | tee <fname> Output is printed to screen and to file <fname>

With the following construction, the standards output and the standard err output can be written to different files:  
(command > <stdoutfile>) >& <stderrfile>

## 17. Processes

**ps** show my processes  
**ps -ef** show all processes  
**top** show process information interactively  
**kill** <PID> finishes job with id <PID>; you can kill only processes that you own  
command & starts command in background  
CTRL-Z interrupts active job  
**bg** sends an interrupted job to background  
**fg** brings interrupted job to foreground

## 18. bash

### 18.1 Environment Variables

global environment variables are:

**HOME** absolute path of my home directory  
**PATH** which paths are scanned for executables?  
**TERM** which terminal-type do i use?  
**USER** my name  
**DISPLAY** where is output printed?  
**PRINTER** default printer

...

- a variable MYVAR is set with the: MYVAR="xyz"
- it is possible to unset a variable: **unset** MYVAR
- **echo** \$<VAR> shows the content of variable <VAR>
- **set** or **env** shows all environment variables
- **HOME**PATH=\$HOME:~/newpath' adds newpath to the HOME:PATH-variable

### 18.2 Masking Special Characters

via " ", ' ', \

## 19. Other Useful Commands

file <fname>	what kind of file is file <fname>?
which <fname>	which path leads to <fname>?
grep, fgrep	search expression in files
find ...	search for files in directory tree
sort ...	sort lines of a textfile
diff ...	compare two files
wc ...	count characters, words and lines in a textfile
gzip <file>	compress file
gunzip <file>	uncompress file
tar ...	build an archive
vi ...	text editor
emacs ...	text editor