Linux cheatsheet

```
uname
who
whois
pwd
date
cal

ls –d /usr/???
[gC]*
{1..10}

alias lh=”ls –lh”

man passwd / man -k SHA1
info passwd
wheris bzip2
which bzip2
type echo / type ls

ls / ls –la / ls –lR / ls –ld
ln / ln -s
du –h
head / head –n-1 / head –n5
tail
cat
less
wc –l
nl
cut –d : -f 2
sort
```

```
uniq
touch

grep / grep –vE ‘^ii.*’ / grep –i
grep –E “1[0-9]{2}-[0-9]{3}” file | cut –d : -f 2

free
ps –aux
ps –fax
pstree
top
uptime
jobs

apt-get update / apt-get upgrade
apt-get install appname
dpkg –l / dpkg –search / dpkg –s

ip addr show
env / printenv

chown
chmod / chmod u=rwx,g=rx,o= /home/user /
chmod 750 /home/user

> file.txt
>> file.txt
ls –la /usr/bin &> file.txt
ls –la /usr/bin 2> file.txt
ls –la /usr/bin > file.txt 2>&1
dd if=/dev/zero of=/home/user/f4_4 bs=1024
count=2000
```

```
kill –SIGTERM 10492
kill –SIGKILL 10492
killall –SIGKILL processname

lsblk
fdisk –l
mkfs / mkfs.ext4 / mkfs.ntfs
mount
umount
lsof
cat /proc/partitions
df –h
mkswap
swapon
swapoff
cat /proc/swaps
/etc/fstab
dd if=/dev/zero of=/dev/sdb1 bs=1024
count=10
fsck

/dev/zero
/dev/random
/dev/null

find . –type d – name “example”
find . –type f –iname “example.*”
find . –not –type f –iname “example.*”
find . –name “abc*” ! –name “*.php”
find . –name “*.php” –o –name “*.txt”
find . –type f –mmin +1 –mmin -5
find . –size +5M
find . –empty
find . –perm 777
find example/ -type d –exec chmod 775 {}  +
find . –maxdepth 1 –type f –name “*.jpg”
–exec rm {} +

tar –cvf example.tar directory/
tar –tf example.tar
tar –xvf example.tar
gzip example.tar
gunzip example.tar
bzip2 example.tar
bunzip2 example.tar
tar –cvzf example.tar.gz directory/
tar –xvzf example.tar.gz
tar –cvjf example.tar.bz2 directory/
tar –xvjf example.tar.bz2
gzip < /directory/example > example.gz
bzip2 < /directory/example > example.bz2

sha1sum file
```

Regular expressions

Format

^ and $	Start / end of a line
.	Any character
[ ] and [^]	Any character (not) between the brackets
?	Zero or one time previous character / expression
* and +	Zero or more / one time previous character / expression
{x,y}	Minimum x and maximum y previous character / expression
( )	Group

Character classes

\w and \W	“word character” (a-zA-Z_) and inverse
\b and \B	“word boundary” (boundary from a word) and inverse
\s and \S	Whitespace and inverse
[[:alpha:]]	a-zA-Z
[[:digit:]]	0-9
[[:alnum:]]	a-zA-Z0-9
\d and ...	Not in grep: same as [[:digit:]]

Examples

KdG student numbers: [0-9]{7}-[0-9]{2}

Hexadecimal number of 4 numbers: [0-9A-Fa-f]{4}

Each number containing minimum 3 zeros, repeated after each other:
[0-9]*0{3}[0-9]*

Word “fix” in a text, different possibilities:
[[:space:]]fix[[:space:]]
fix\W
\<fix\>

Start with <, contains @ and ends with >:
<.+@.+>