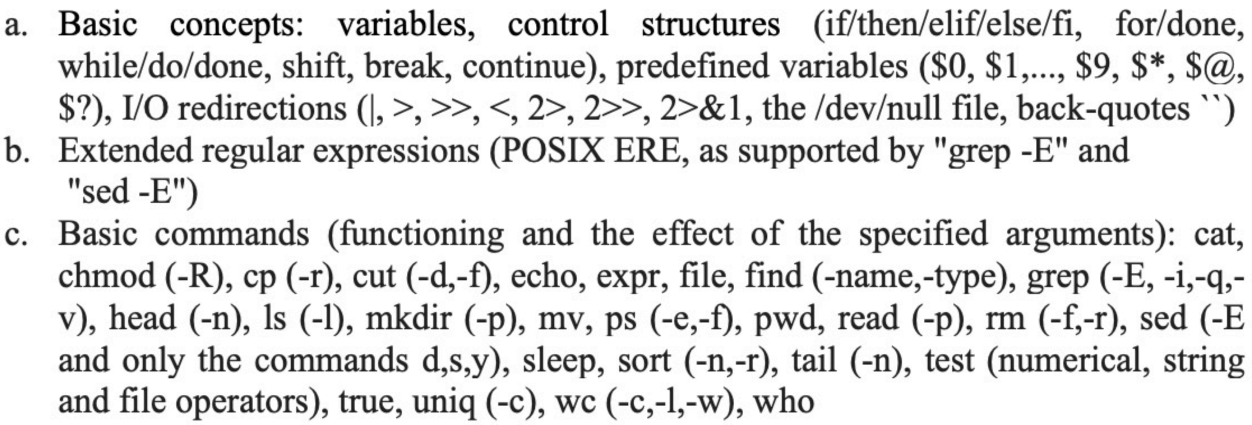
> ls -l

argument, option that modifies the way the comands work

> ls - l . → current directory

> touch a → creates a file



a. Basic concepts: variables, control structures (if/then/elif/else/fi, for/done, while/do/done, shift, break, continue), predefined variables ($0, $1,..., $9, $*, $@, $?), I/O redirections (|, >, >>, <, 2>, 2>>, 2>&1, the /dev/null file, back-quotes ``)

b. Extended regular expressions (POSIX ERE, as supported by "grep -E" and "sed -E")

c. Basic commands (functioning and the effect of the specified arguments): cat, chmod (-R), cp (-r), cut (-d,-f), echo, expr, file, find (-name,-type), grep (-E, -i,-q,-v), head (-n), ls (-l), mkdir (-p), mv, ps (-e,-f), pwd, read (-p), rm (-f,-r), sed (-E and only the commands d,s,y), sleep, sort (-n,-r), tail (-n), test (numerical, string and file operators), true, uniq (-c), wc (-c,-l,-w), who

> pwd → print working directory

> cd ..

goes to parent directory

* the only exception is when you're in the root folder '/'

> cd → goes to the user's home directory

grep = a test search tool (returns the line that contain the expression)

> grep -E "i" a

↑ expression ↑ file

> grep -E "i[0-9]" a

↑ match any one of these

^ - caret → beginning of line

> grep -E "^i[0-9]" a

$ → end of the line

* point matches any char ⇒ for special chars you do '\.'

> grep -E "@ubbcluj\.ro" a

> grep -E "^[0-9a-zA-Z]@ubbcluj\.ro" a

just one char

```
> grep -E "^[0-9a-zA-Z]+@ubbcluj\.ro" a
```

repeat one or more times

```
> grep -E "^[0-9a-zA-Z][0-9a-zA-Z.]*@ubbcluj\.ro" a
```

repeat 0 or more times

+ → {1, }

* → {0, }

```
> grep -E -c ".*" /dir/...
```

same as

```
> wc /dir/...
```

```
> grep -E -v "[aeiou]" a
```

-v : reverse match

same as

```
> grep -E "[^aeiou]" a
```

-q option stands for quiet ⇒ doesn't print output

if it finds smt it returns 0, if not, 1

```
echo $?
```

<u>sed</u> → stream editor

* commonly used for search and replace

```
> sed -E "s/ubbcluj.ro$/gmail.com/" a
```

but this is case sensitive

sed, by default, gives a copy of the file

-i for sed is not case insensitive, but rather modifies the fil

```
> sed -E "s/ubbcluj.ro$/gmail.com/i" a
```

i : case insensitive

* sed replaces only the first occurrence

```
> sed -e "s/ubbcluj.ro$/gmail.com/g" a
```

↑ global it changes all occurrences

```
> sed -e "s/([0-9])/-- \1 --/g" a
```

↑ you can make a reff. to them

↳ every digit now will be replaced as such

```
> sed -e "s/([aeiou])/\1\1/g" a
```

↳ duplicates vowels

```
> sed -e "s/([aeiou])([aeiou])/\2\1/g" a
```

↳ Swaps 2 consec. vowels

```
> sed -e "y/abc/123/"
```

* transliteration

* restriction is to have both strings of the same lenght

REGEX/d

```
> sed -e "/i/d" a
```

↳ deletes every line that contain i

* to delete a sequence but not remove the whole line

```
> sed -e "s/ubbcluj\.ro//" file
```

↳ search and replace with the empty string

awk

```
> awk 'condition {actions}'
```

NR - line number

```
> awk 'NR == 1 {print "text"}' a
> awk 'NR == 1 {print $0}' file
```

↳ prints the content of first line

$0 prints the whole line

```
> awk -F: 'NR>0 && NR<10 {print $1, $2, $3}' file
```

↑ use any field sep :

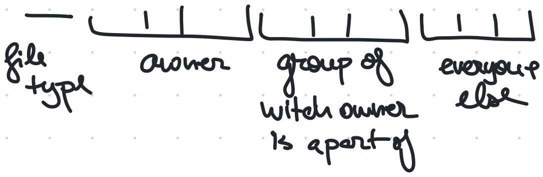
```
> awk   -F:   'BEGIN {s=0} {s=s+1} END {print s}' file
```

~ matches a regular expression

```
> awk   -F|   'BEGIN {s=0}  $1 ~ /^ex[0-9]+/ {s=s+1} END {print s}' file
```

FNR → current counter within current file

permissions



chmod   u-r, o+x

user
revoke read rights
other
give execution rights

chmod   u=rwx, g-wx, o-wx   my-exe

or use binary

chmod   755   my-exe

7: 1 1 1
(R W X)

5: 1 0 1
(R – X)

the operathing systen executes sut that it has a path to

./ → refers to current directory

if you move the exeatable into user/bin you can run with just the name

$\boxed{\$ \text{PATH}}$