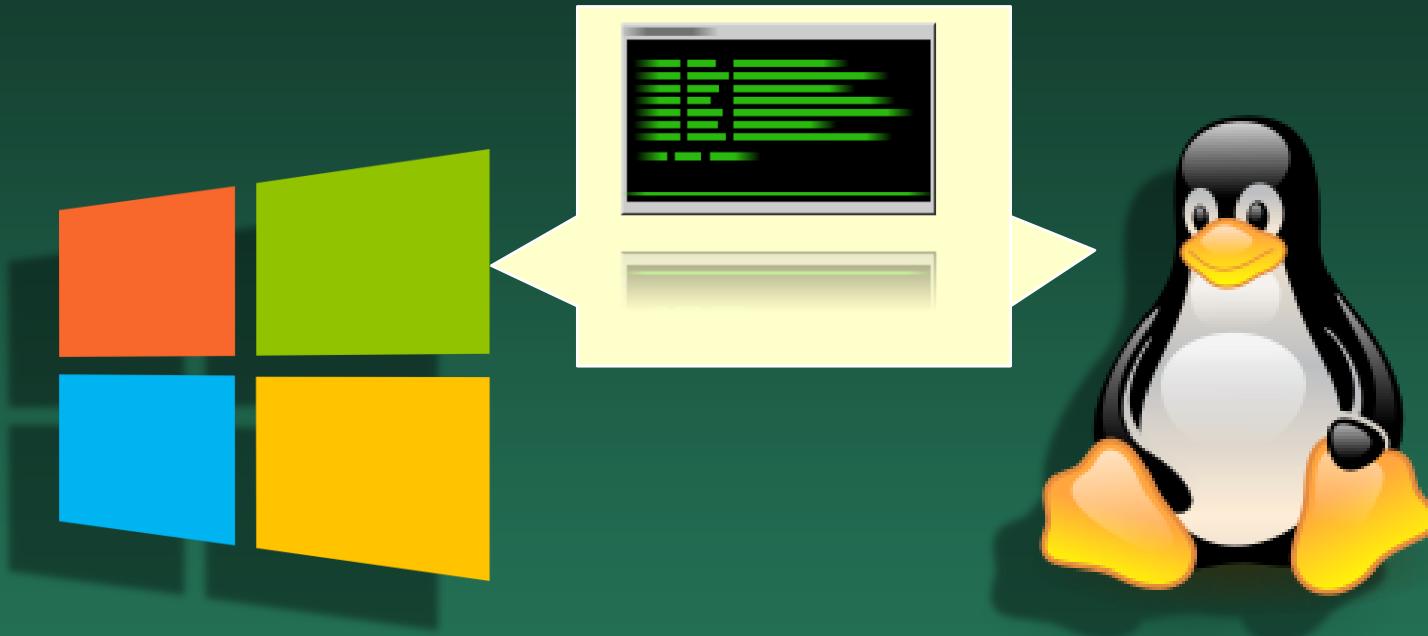


# System shells



## BASH syntax - 6

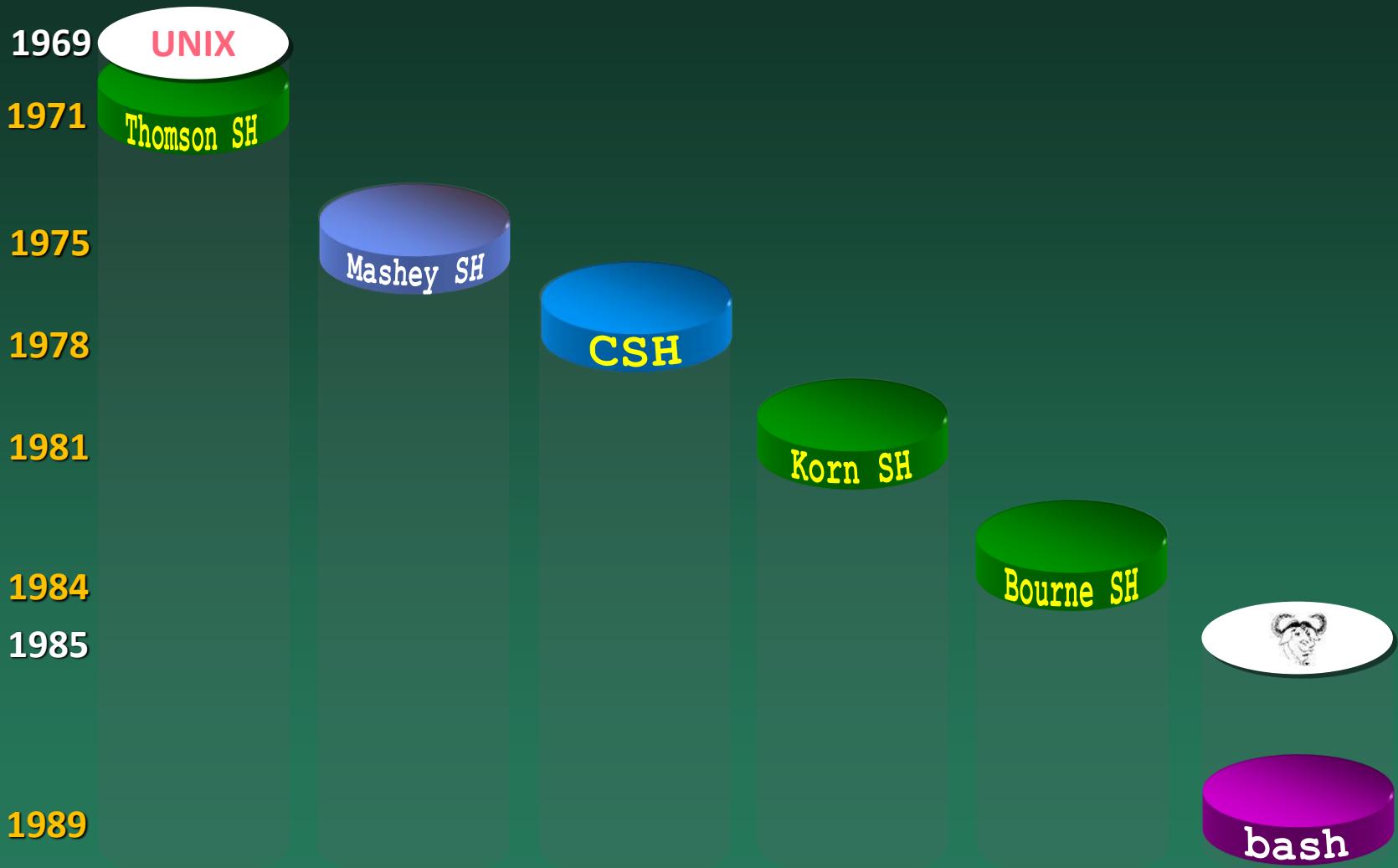


BASH

# 1 - BASH CMD

CD, MKDIR, RM .

# 18 years = Majority



# BASH Command line

## ◆ A long history of command shells

- sh, ksh, ash, bash, dash, ...
- More functions... better efficiency...
- All are the /bin/sh program
- ls -l /bin/sh gives the link

## ◆ Inherited from Unix (!DOS)

- Case sensitive
  - foo.exe != FOO.EXE != Foo.exe
  - cd is a command, CD is not
- NO Volume letter prefix ( /usr, /etc/bin etc.)
- Separator symbol = /

# BASH naming limitations

## ◆ Forbidden character in **UNIX** names

- / slash = namespace separator
- <NewLine> = display separator



## ◆ **BASH** special characters <> | & ; \* ? ( ) " ' ` \\$ { } [ ]



## ◆ + **Separators** <Space> , <TAB> and <NewLine>

- All can be escaped with \
- Or deactivated inside quotes ' ... ' *excluding*'
- Or deactivated inside dbl-quotes " ... " *excluding* " ` \\$

- Possible file names:

```
'mon fichier.txt'  
"my'file.txt"  
'my>file.txt'  
'my?file.txt'
```



mon fichier.txt  
my'file.txt  
my>file.txt  
my?file.txt

→ *BEWARE: file & folder names with BASH characters may cause bugs in « unaware » scripts*

# BASH commands list

## ◆ Type **help** and **<command> --help**

```
uba64@TWEETY:/mnt/c/unix  x  +  v  -  □  ×
break [n]
builtin [shell-builtin [arg ...]]
caller [expr]
case WORD in [PATTERN [| PATTERN]...) COMMAND>
cd [-L|[-P [-e]] [-@]] [dir]
command [-pVv] command [arg ...]
compgen [-abcdefgjksuv] [-o option] [-A actio>
complete [-abcdefgjksuv] [-pr] [-DEI] [-o opt>
compopt [-o]+o option] [-DEI] [name ...]
continue [n]
coproc [NAME] command [redirections]
declare [-aAfFgi]nrtux] [-p] [name[=value] ...>
dirs [-c]pv] [+N] [-N]
disown [-h] [-ar] [jobspec ... | pid ...]
echo [-neE] [arg ...]
enable [-a] [-dnps] [-f filename] [name ...]
eval [arg ...]
exec [-cl] [-a name] [command [arguments ...]>
exit [n]
export [-fn] [name[=value] ...] or export -p
false
fc [-e ename] [-lnr] [first] [last] or fc -s >
fg [job_spec]
for NAME [in WORDS ... ] ; do COMMANDS; done
for (( exp1; exp2; exp3 )); do COMMANDS; don>
function name { COMMANDS ; } or name () { COM>
getopts optstring name [arg]
hash [-lr] [-p pathname] [-dt] [name ...]
help [-dms] [pattern ...]
uba64@TWEETY:/mnt/c/unix$ bash -c help
printf [-v var] format [arguments]
pushd [-n] [+N | -N | dir]
pwd [-LP]
read [-ers] [-a array] [-d delim] [-i text] >
readarray [-d delim] [-n count] [-o origin] >
readonly [-aAf] [name[=value] ...] or readonly
return [n]
select NAME [in WORDS ... ;] do COMMANDS; do>
set [-abefhkmnptuvxBCHP] [-o option-name] [->
shift [n]
shopt [-pqsu] [-o] [optname ...]
source filename [arguments]
suspend [-f]
test [expr]
time [-p] pipeline
times
trap [-lp] [[arg] signal_spec ...]
true
type [-afptP] name [name ...]
typeset [-aAfFgi]nrtux] [-p] name[=value] .>
ulimit [-SHabcdefiklmnpqrstuvxPT] [limit]
umask [-p] [-S] [mode]
unalias [-a] name [name ...]
unset [-f] [-v] [-n] [name ...]
until COMMANDS; do COMMANDS; done
variables - Names and meanings of some shell>
wait [-fn] [id ...]
while COMMANDS; do COMMANDS; done
{ COMMANDS ; }
```

# The UNIX manual

## ◆ Type **man <command>**

CP(1) User Commands CP(1)

### NAME

**cp** - copy files and directories

### SYNOPSIS

**cp** [OPTION]... [-T] SOURCE DEST  
**cp** [OPTION]... SOURCE... DIRECTORY  
**cp** [OPTION]... -t DIRECTORY SOURCE...

### DESCRIPTION

Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.

#### **-a, --archive**

same as **-dR --preserve=all**

#### **--attributes-only**

don't copy the file data, just the attributes

#### **--backup [=CONTROL]**

make a backup of each existing destination file

# Choose your colors

## ◆ Terminal colors

White: [ \033[0;37m ]  
Cyan: [ \033[0;36m ]  
Magenta: [ \033[0;35m ]  
Blue: [ \033[0;34m ]  
Yellow: [ \033[0;33m ]  
Green: [ \033[0;32m ]  
Red: [ \033[0;31m ]  
Black: [ \033[0;30m ]

bold White: [ \033[1;37m ]  
...etc...

## ◆ Prompt is PS1 variable

```
PS1='\[ \033[0;32m\] \w\]> '
```

```
ubu64@TWEETY:/mnt/c/unix$ pwd  
/mnt/c/unix  
ubu64@TWEETY:/mnt/c/unix$ echo $PS1  
\[\e[0;\u@h: \w\a\]\${debian_chroot:+($debian_chroot)}\[ \033[01;32m\]\u@h\[ \033[00m\]  
ubu64@TWEETY:/mnt/c/unix$ PS1='\[ \033[0;31m\]\w\]> \[ \033[1;36m\]'  
/mnt/c/unix> pwd  
/mnt/c/unix  
/mnt/c/unix>
```

# Basic commands

## ◆ Clear the screen

**clear**

## ◆ Where am I ?

**pwd**

## ◆ Folder content

- ...Simple list
- Detailed list

**ls**

**ls -al**

```
/mnt/c/unix\> ls -al
total 2482
drwxrwxrwx 1 ubu64 ubu64    512 Feb 24 10:17 .
drwxrwxrwx 1 ubu64 ubu64    512 Feb 22 20:34 ..
drwxrwxrwx 1 ubu64 ubu64    512 Aug 30 12:53 2022 S1
-rwxrwxrwx 1 ubu64 ubu64   61952 Oct 21 12:28 4M2301.exe
drwxrwxrwx 1 ubu64 ubu64    512 Jan  6 17:23 Driver
drwxrwxrwx 1 ubu64 ubu64    512 Mar  6 2022 EasyBMP
drwxrwxrwx 1 ubu64 ubu64    512 Jul  7 2022 FREERTOS_SIMU_WIN_POSIX
drwxrwxrwx 1 ubu64 ubu64    512 Sep  7 16:34 Posix_GCC
-rwxrwxrwx 1 ubu64 ubu64     35 Nov 29 11:51 bb
-rwxrwxrwx 1 ubu64 ubu64   17120 Dec  5 13:37 goal
-rwxrwxrwx 1 ubu64 ubu64    1163 Dec  5 14:28 goal.c
-rwxrwxrwx 1 ubu64 ubu64   2768 Dec  5 13:34 goal.o
```

# Pattern matching for names

## ◆ Generalize names with *metacharacters*

- \* Zero or any number of any character
- ? One of any character
- [class] One character in the group i.e [Aez12]
- [^class] One character NOT in the group [^ezEZ]
- [x-y] One character in the interval [0-9]
- \ Escape for literal usage of a metacharacter \\*

## ◆ Works with **ls**, **rm**, **mv**, **cp**, etc.

```
/mnt/c/unix/SH_files> ls  
baba.txt subsub tata.txt tjpr zaza.txt  
/mnt/c/unix/SH_files> ls [^tz]a???.txt  
baba.txt  
/mnt/c/unix/SH_files> ls t*.*  
tata.txt  
/mnt/c/unix/SH_files> ls t*  
tata.txt tjpr
```

→ Looks like *REGEX* but it's *NOT REGEX*

# File exploration

## ◆ Content of a file                            **file**

- Displays the nature of the file content

```
/mnt/c/unix/SH_files\> file *
baba.txt: ASCII text
goal.c: C source, ASCII text, with CRLF line terminators
goal.o: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), not stripped
subsub: directory
tata.txt: ASCII text
tjpr: Bourne-Again shell script, ASCII text executable
zaza.txt: ASCII text
```

- Not only the file extension

```
/mnt/c/unix/SH_files\> mv goal.o goal.txt
/mnt/c/unix/SH_files\> file g*
goal.c: C source, ASCII text, with CRLF line terminators
goal.txt: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), not stripped
```

# Deep file exploration

- ◆ Detailed content of a file **od**
  - ...Octal dump
- ◆ Numerous display formatting options
  - Octal, Hexa, Decimal
  - ASCII
  - Big-Endian, Little-Endian
  - Options may be combined

```
/mnt/c/unix/SH_files\> od -ct x1 bash.txt
0000000  I   K   N   O   W   M   Y
          49  20  4b  4e  4f  57  20  4d  59  20  42  41  53  48  0a
0000017
```

# File dump



## ◆ Format options

- C include style compatibility
  - Little-endian output

```
/mnt/c/unix/SH_files> head --bytes=54 arr1.bmp | xxd -i
0x42, 0x4d, 0x3a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x36, 0x00,
0x00, 0x00, 0x28, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00,
0x00, 0x00, 0x01, 0x00, 0x18, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

# Command history

- ◆ Previous commands **history**
- ◆ To repeat a command in the history
  - ! Followed by the command index in history
  - ! Followed by the first letter of the command in history
  - !! To repeat the last command
- ◆ The commands history is also available with <UP> and <DOWN> keys
- ◆ A command may be edited before re-execution

# Navigate the File system

- ◆ Move to a folder      `cd /bin`
- ◆ Identify files      `ls -l bash`
- ◆ Go back home      `cd ~`

```
/mnt/c/unix/SH_files> pwd  
/mnt/c/unix/SH_files  
/mnt/c/unix/SH_files>
```

Where am I ? = PROMPT  
*OR*  
Type `pwd`

# Folder control

- ◆ Create a sub-folder

`mkdir AA`

- ◆ Create a folder

`mkdir /AA/BB`

- ◆ Remove a sub-folder

- Removed if empty

`rmdir AA`

- ◆ Remove hierarchy

- Remove BB and AA

`rmdir /p /AA/BB`

- ◆ Remove empty folders

`rm /r /d`

- ◆ Remove everything

`rm /r`

# Special folder names

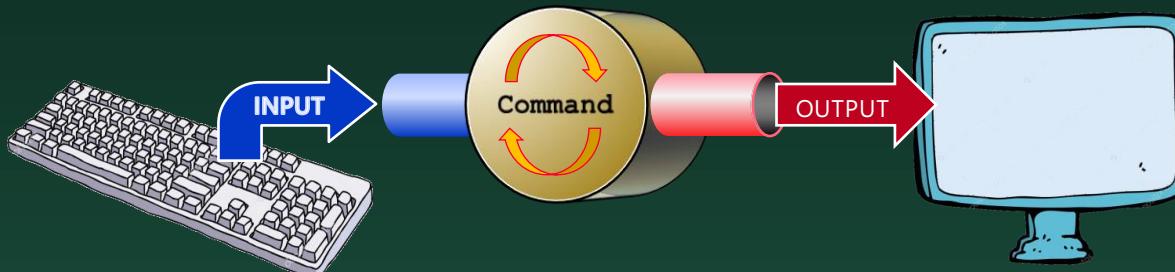
## ◆ Type **ls -al**

- Shows " ." And " .. " Folders
- . Is the current folder
- .. Is the parent folder of .

## ◆ Type **CD ..\..**

- Brings you two levels back
- Can be used in any command
- Can be used in any path ..\..\AA\..
- You cannot create other file/folder " ." and " .. "

# I/O Redirections



## ◆ Standard INPUT is keyboard

- May be redirected with <

`command < info.dat`

## ◆ Standard OUTPUT is screen (TTY)

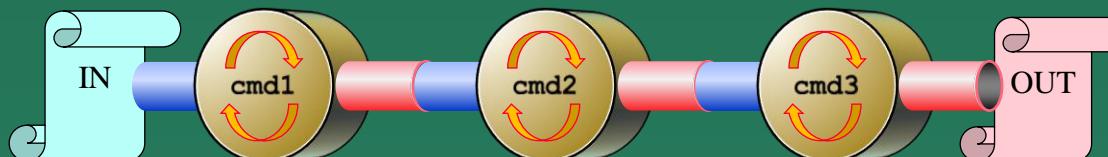
- May be redirected with > or >>

`command >> resu.dat`

## ◆ Make a flow of commands (Pipe)

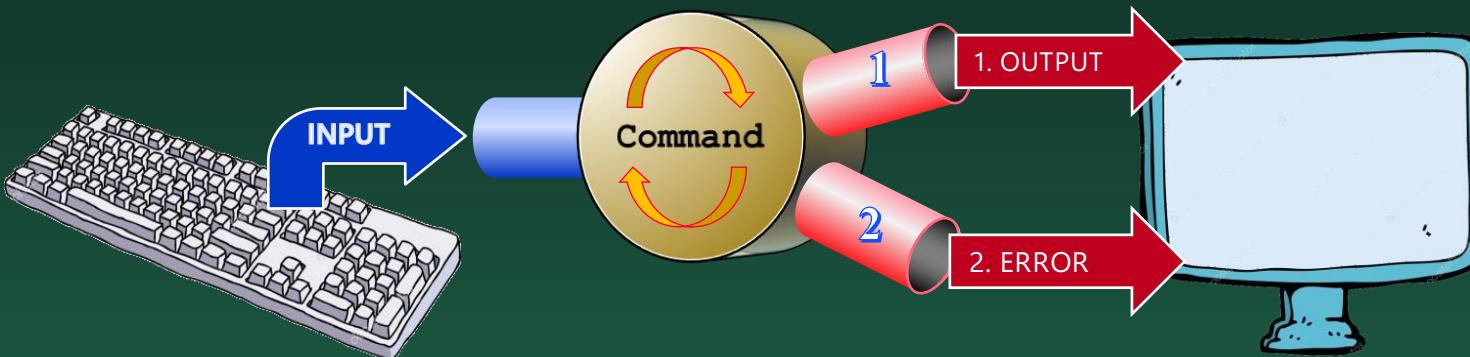
- Connect OUTPUT to INPUT with |

`command1 | command2`



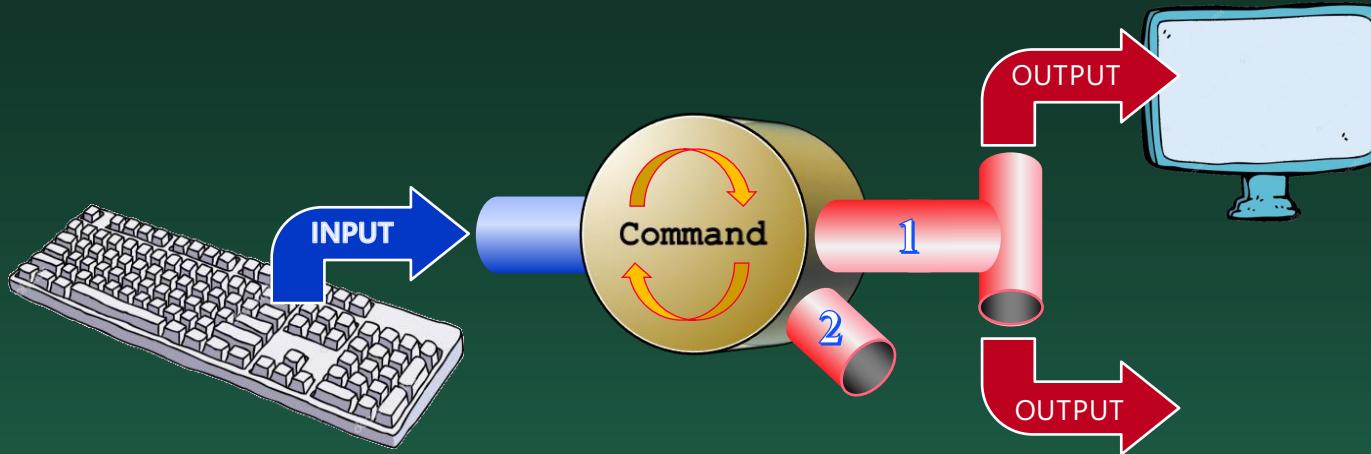
`cat in.txt > cmd1 | cmd2 | cmd3 > out.txt`

# Special case of output



- ◆ **OUTPUT** is made of TWO channels
  - 1=STANDARD & 2=ERROR
  - ERROR may be redirected with **2>** or **2>>** (*> means 1>*)
  - Command **2> error.log**
  - Command **&> alloutputs.log** *redirects 1 & 2*
- ◆ Programs must send their error messages to ERROR
  - Not always followed ☹

# Split standard output



## ◆ Command **tee** is a splitter      `command | tee data.txt`

- Keeps TTY output and adds a file stream
- `Command | tee actions.log`                    *no need for redirection*
- `Command | tee -a allactions.log`            *tee appends to file*

## ◆ Use **tee** on ERRORS with redirection

- `/dev/null` = nowhere                            `command 2>&1 | tee /dev/null`

# BATCH is talking

- ◆ Type **echo I KNOW MY BATCH**
- ◆ Will treat '**'** and "**"** as delimiters
- ◆ Special characters must be escaped

```
/mnt/c/unix/SH_files> echo I KNOW MY BASH  
I KNOW MY BASH  
/mnt/c/unix/SH_files> echo 'I KNOW MY BASH'  
I KNOW MY BASH  
/mnt/c/unix/SH_files> echo "I KNOW MY BASH"  
I KNOW MY BASH  
/mnt/c/unix/SH_files> echo $?/(  
-bash: syntax error near unexpected token `('  
/mnt/c/unix/SH_files> echo \$\?\\/(  
$?/(
```

# Create files

## ◆ Create a file with content

- `echo I KNOW MY BASH > bash.txt`
- `ls *.txt`

## ◆ Add content to a file

- `echo ONE MORE LINE >> bash.txt`

## ◆ Text editors

- `vi bash.txt`
- `nano mychoice.txt`

edit mode / cmd mode ☹  
cmd menu ☺

# Read files

## ◆ Display file content

- `cat bash.txt`

## ◆ Concatenate files

- `cat bash.txt bash.txt > babash.txt`
- Accept several file names as arguments

## ◆ File head

- `head -n 20 bigfile.txt`

## ◆ File tail

- `tail -n 20 bigfile.txt`

# Manipulate files/folders

## ◆ Make a copy of a file/folder

- `cp bash.txt babash.txt`
- `mkdir subfolder`
- `cp bash.txt subfolder`
- `cp -r subfolder anotherfolder`

## ◆ Transport a file/folder

- `mv bash.txt babash.txt`
- `mkdir subfolder`
- `mv babash.txt subfolder`
- `mv subfolder anotherfolder`

# Delete files

## ◆ Delete a file/folder

- `cp bash.txt subfolder`
- `rm bash.txt`
- `rm -r subfolder`
- BEWARE!! `rm` has no UNDO (*better use -i*)

## ◆ Eliminate bytes on the storage

- `shred bash.txt`

# Misc. commands

- ◆ Actual date & time **date**
  - ◆ Terminates CMD **exit**
  - ◆ Terminate the session **logout**
  - ◆ Power down the PC **shutdown**
  - ◆ Running processes **ps**
  - ◆ Version of the kernel **uname -r**
  - ◆ System & Hardware **uname -a**
  - ◆ Disk usage **du -s**
  - ◆ Storage free space **df**

# Control the scroll

- ◆ Type **help | more** Shows one screen at a time
  - <ENTER> scroll one (*more*) line
  - <SPACE> scroll one (*more*) screen
  - Q end *more* command
- ◆ Type **help > shelp.txt**
- ◆ Type **more < shelp.txt**
- ◆ Type **cat shelp.txt | more**
  
- ◆ Type **less shelp.txt**
  - More scrolling options ☺

# Parameter separators

- ◆ Type **mkdir AA BB CC DD**
  - Will create folders **AA**, **BB**, **CC** and **DD**
- ◆ Type **rmdir AA BB CC DD**
  - Will remove folders
- ◆ Type **mkdir "AA BB CC DD"**
  - No interpretation inside " "
  - Creates folder with long name **AA BB CC DD**

# Commands on multiple lines

## ◆ Type **(md AA**

- Will not execute command
- Asks for more information until we type **)**
- May be nested

## ◆ To split a command on multiple lines

- Escape **<EOL>** with \
- Type **(mkdir \ ← beware the space separator**
- Type **AA \ ← beware the space separator**
- Type **BB\**
- Type **CC) ← will create folder "BBCC"**

# Multiple commands

- ◆ Separate commands with " ; "  
• Group commands with " { } "

`cmd1 ; cmd2`  
`{ cmd1 ; cmd2 ; }`

## ◆ OR Conditional command operators

- Operator " || " will filter errors  
• cmd2 executes if cmd1 fails

`cmd1 || cmd2`

## ◆ AND Conditional command operators

- Operator " && " will chain  
• cmd2 only executes if cmd1 succeeded

`cmd1 && cmd2`