OS Lab

Eng. Yousefnezhad

OperatingSystem@Lab:~$ Github

This file is also on GitHub

OperatingSystem@Lab:~$ Github

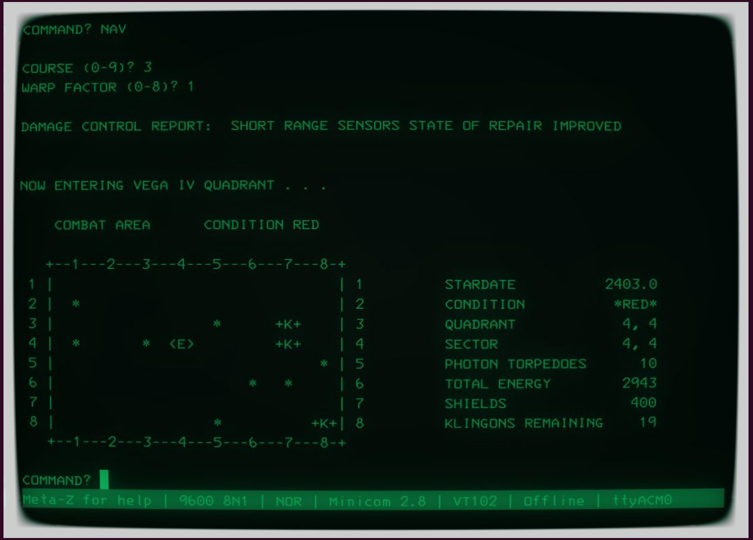
This file is also on GitHub

OperatingSystem@Lab:~$ clear

OperatingSystem@Lab:~$ Linux

Older Unix versions had no GUI. They worked only through terminal!

So it makes sense that many commands must exist to work and navigate on Linux.



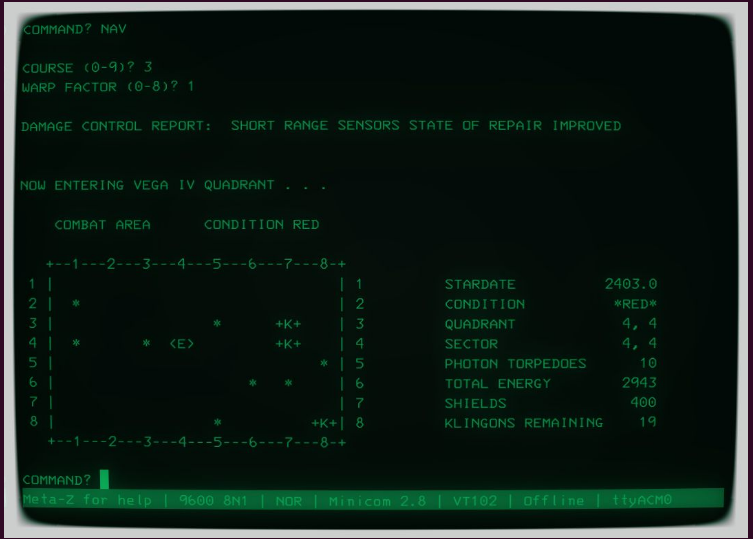
```
COMMAND? NAV

COURSE (0-9)? 3
WARP FACTOR (0-8)? 1

DAMAGE CONTROL REPORT:  SHORT RANGE SENSORS STATE OF REPAIR IMPROVED


NOW ENTERING VEGA IV QUADRANT . . .

    COMBAT AREA      CONDITION RED

  +--1---2---3---4---5---6---7---8-+
1 |                                | 1        STARDATE            2403.0
2 |  *                             | 2        CONDITION            *RED*
3 |                  *      +K+    | 3        QUADRANT              4, 4
4 |  *       *  <E>         +K+    | 4        SECTOR                4, 4
5 |                              * | 5        PHOTON TORPEDOES        10
6 |                      *   *     | 6        TOTAL ENERGY          2943
7 |                                | 7        SHIELDS                400
8 |                  *          +K+| 8        KLINGONS REMAINING      19
  +--1---2---3---4---5---6---7---8-+

COMMAND? 
Meta-Z for help | 9600 8N1 | NOR | Minicom 2.8 | VT102 | Offline | ttyACM0
```

old Linux versions

OperatingSystem@Lab:~$ Linux

Older Unix versions had no GUI. They worked only through terminal!

So it makes sense that many commands must exist to work and navigate on Linux.



```
COMMAND? NAV

COURSE (0-9)? 3
WARP FACTOR (0-8)? 1

DAMAGE CONTROL REPORT:  SHORT RANGE SENSORS STATE OF REPAIR IMPROVED


NOW ENTERING VEGA IV QUADRANT . . .

    COMBAT AREA       CONDITION RED

  +--1---2---3---4---5---6---7---8-+
1 |                                | 1          STARDATE              2403.0
2 |  *                             | 2          CONDITION              *RED*
3 |                  *      +K+    | 3          QUADRANT                4, 4
4 |  *       *  <E>         +K+    | 4          SECTOR                  4, 4
5 |                              * | 5          PHOTON TORPEDOES          10
6 |                      *   *     | 6          TOTAL ENERGY            2943
7 |                                | 7          SHIELDS                  400
8 |                  *          +K+| 8          KLINGONS REMAINING        19
  +--1---2---3---4---5---6---7---8-+

COMMAND? 
Meta-Z for help | 9600 8N1 | NOR | Minicom 2.8 | VT102 | Offline | ttyACM0
```

old Linux versions

OperatingSystem@Lab:~$ clear

OperatingSystem@Lab:~$ Commands --help

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ cd /your/dir/path/
```

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ cd /your/dir/path/

OperatingSystem@Lab:~$ pwd
```

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ cd /your/dir/path/
```

```
OperatingSystem@Lab:~$ pwd
```

```
OperatingSystem@Lab:~$ mkdir /path/YourDirName/
```

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ cd /your/dir/path/

OperatingSystem@Lab:~$ pwd

OperatingSystem@Lab:~$ mkdir /path/YourDirName/

OperatingSystem@Lab:~$ ls
```

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ cd /your/dir/path/

OperatingSystem@Lab:~$ pwd

OperatingSystem@Lab:~$ mkdir /path/YourDirName/

OperatingSystem@Lab:~$ ls

OperatingSystem@Lab:~$ touch /path/YourFileName/
```

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ cd /your/dir/path/

OperatingSystem@Lab:~$ pwd

OperatingSystem@Lab:~$ mkdir /path/YourDirName/

OperatingSystem@Lab:~$ ls

OperatingSystem@Lab:~$ touch /path/YourFileName/

OperatingSystem@Lab:~$ clear
```

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ echo "Hello Linux!" >> /path/YourFil
eName/
```

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ echo "Hello Linux!" >> /path/YourFil
eName/

OperatingSystem@Lab:~$ cat /path/YourFlieName/
```

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ echo "Hello Linux!" >> /path/YourFil
eName/

OperatingSystem@Lab:~$ cat /path/YourFlieName/

OperatingSystem@Lab:~$ lshw
```

```
OperatingSystem@Lab:~$ Commands --help
```

In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ echo "Hello Linux!" >> /path/YourFil

eName/

OperatingSystem@Lab:~$ cat /path/YourFlieName/

OperatingSystem@Lab:~$ lshw

OperatingSystem@Lab:~$ lsblk
```

```
OperatingSystem@Lab:~$ Commands --help
```

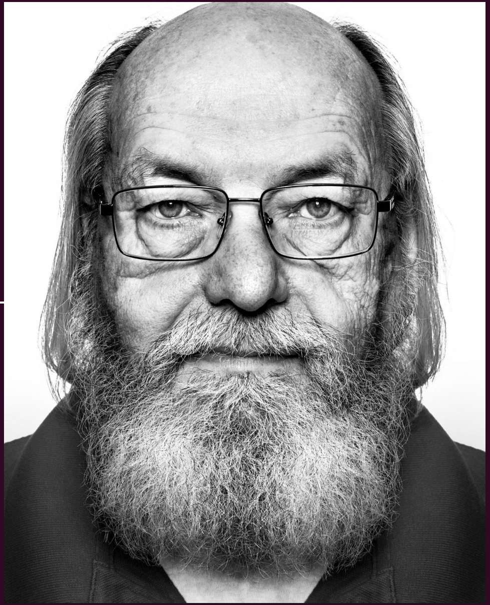
In the last session, we were introduced to a series of popular and widely used commands.

```
OperatingSystem@Lab:~$ echo "Hello Linux!" >> /path/YourFil

eName/

OperatingSystem@Lab:~$ cat /path/YourFlieName/

OperatingSystem@Lab:~$ lshw

OperatingSystem@Lab:~$ lsblk

OperatingSystem@Lab:~$ clear
```

OperatingSystem@Lab:~$ Thompson-shell

The Thompson shell was the first Unix shell, and was written by Ken Thompson.

It was a simple command interpreter, not designed for scripting, but nonetheless introduced several innovative features to the command-line interface and led to the development of the later Unix shells.

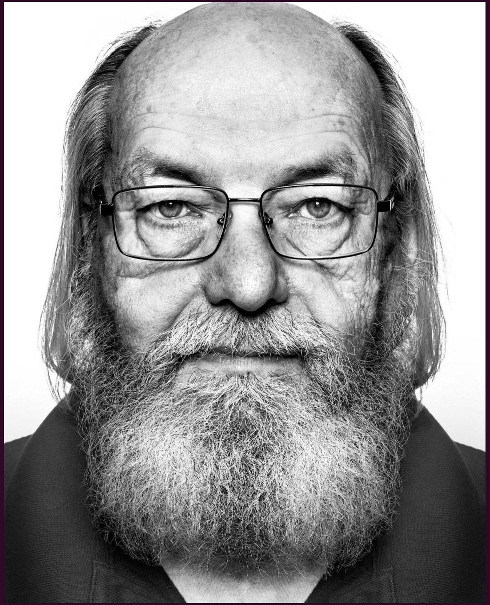

Ken Thompson

```
OperatingSystem@Lab:~$ Thompson-shell
```

The Thompson shell was the first Unix shell, and was written by Ken Thompson.

It was a simple command interpreter, not designed for scripting, but nonetheless introduced several innovative features to the command-line interface and led to the development of the later Unix shells.

```
OperatingSystem@Lab:~$ clear
```


Ken Thompson

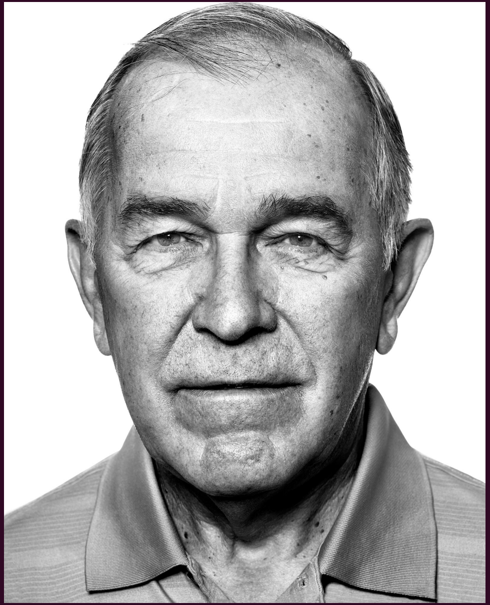
OperatingSystem@Lab:~$ Bourne-shell

Created by Stephen Bourne for V7 UNIX.
It remains a useful shell today (in some cases, as the default root shell).

Bourne introduced:

- control flows - variables into scripts
- loops - providing a more functional language to interact with the OS

But the shell lacked the ability to define functions. (ಥ﹏ಥ)


Steve Bourne

OperatingSystem@Lab:~$ Bourne-shell

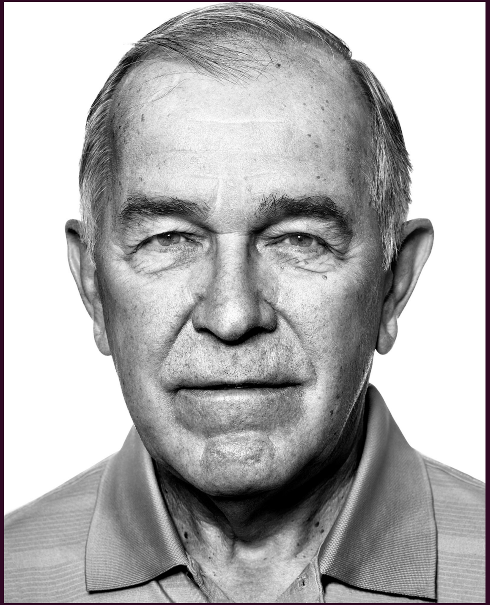
Created by Stephen Bourne for V7 UNIX.
It remains a useful shell today (in some cases, as the default root shell).

Bourne introduced:

- control flows
- loops
- variables into scripts
- providing a more functional language to interact with the OS

But the shell lacked the ability to define functions. (ಥ﹏ಥ)

OperatingSystem@Lab:~$ clear


Steve Bourne

OperatingSystem@Lab:~$ Bourne-Again --SHell

What we use today is actually a descendant of Bourne Shell, which underwent changes over time that made it better.
BASH Created in 1989 by Brian Fox for the GNU Project, and designed as a 100% free alternative for the Bourne shell (sh) and other proprietary Unix shells.
The keywords, syntax, dynamically scoped variables, and other basic features of the language are all copied from the Bourne shell, (sh). Other features, e.g., history, are copied from the C shell, (csh), and the Korn Shell, (ksh).

OperatingSystem@Lab:~$ Bourne-Again --SHell

What we use today is actually a descendant of Bourne Shell, which underwent changes over time that made it better.
BASH Created in 1989 by Brian Fox for the GNU Project, and designed as a 100% free alternative for the Bourne shell (sh) and other proprietary Unix shells.
The keywords, syntax, dynamically scoped variables, and other basic features of the language are all copied from the Bourne shell, (sh). Other features, e.g., history, are copied from the C shell, (csh), and the Korn Shell, (ksh).

OperatingSystem@Lab:~$ clear

OperatingSystem@Lab:~$ bash-script

Today, with the changes that have occurred, we can program with the same commands.

But we want to talk about a few more things. For example, shebang, alias and cronjob.

```
OperatingSystem@Lab:~$ bash-script
```

Today, with the changes that have occurred, we can program with the same commands.

But we want to talk about a few more things. For example, shebang, alias and cronjob.

```
OperatingSystem@Lab:~$ clear
```

```
OperatingSystem@Lab:~$ cat random_bash.sh
#!/bin/bash

name="John"
echo "Hello $name!"
```

```
OperatingSystem@Lab:~$ cat random_bash.sh
#!/bin/bash

name="John"
echo "Hello $name!"
OperatingSystem@Lab:~$ what is #!
```

This #! is called shebang or hashbang.
Shebang has a special meaning when it is used in the very first line of the script.
It is used to specify the interpreter with which the given script will be run by default.

```
echo "Hello $name!"
```

```
OperatingSystem@Lab:~$ what is #!
```

This #! is called shebang or hashbang.
Shebang has a special meaning when it is used in the very first line of the script.
It is used to specify the interpreter with which the given script will be run by default.

```
OperatingSystem@Lab:~$ shebang -example
```

#!/bin/bash → means the interpreter should be bash shell.
#!/bin/zsh → means the interpreter to be used is Z shell.

```
name="John"
echo "Hello $name!"
```

OperatingSystem@Lab:~$ what is #!

This #! is called shebang or hashbang.
Shebang has a special meaning when it is used in the very first line of the script.
It is used to specify the interpreter with which the given script will be run by default.

OperatingSystem@Lab:~$ shebang -example

#!/bin/bash → means the interpreter should be bash shell.
#!/bin/zsh → means the interpreter to be used is Z shell.

OperatingSystem@Lab:~$ clear

OperatingSystem@Lab:~$ alias

BASH Alias is a shortcut to run commands using some mapping.
The alias keyword replaces the command with the string which might be sets of commands or functions.
The alias is defined in the ~/.bashrc or ~/.bash_profile.
These files are loaded in the shell environment and thus the commands listed in the alias are also been loaded and ready to be executed.

```
OperatingSystem@Lab:~$ alias
```

BASH Alias is a shortcut to run commands using some mapping.
The alias keyword replaces the command with the string which might be sets of commands or functions.
The alias is defined in the ~/.bashrc or ~/.bash_profile.
These files are loaded in the shell environment and thus the commands listed in the alias are also been loaded and ready to be executed.

```
OperatingSystem@Lab:~$ clear
```

```
OperatingSystem@Lab:~$ alias -OwnMade

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Python3 alias
alias python='python3'

# 'mkdir' + 'cd' alias
function mkcdir() {
    mkdir -p "$1"
    cd "$1"
}
```

```
    mkdir -p "$1"
    cd "$1"
}

# cowsay
if [ -x /usr/games/cowsay -a -x /usr/games/fortune ]; then
    cowsay "Hi Amir!" | lolcat
fi

# open directory with GUI alias
alias open='nautilus ./'

# check battery health alias
alias batthealth='upower -i /org/freedesktop/UPower/devices/battery_BAT0'
```

```
    mkdir -p "$1"
    cd "$1"
}

# cowsay
if [ -x /usr/games/cowsay -a -x /usr/games/fortune ]; then
    cowsay "Hi Amir!" | lolcat
fi

# open directory with GUI alias
alias open='nautilus ./'

# check battery health alias
alias batthealth='upower -i /org/freedesktop/UPower/devices/battery_BAT0'
```

```
OperatingSystem@Lab:~$ clear
```

OperatingSystem@Lab:~$ cronjob

A cronjob is a Linux command used for scheduling tasks to be executed sometime in the future.

```
OperatingSystem@Lab:~$ cronjob
```

A cronjob is a Linux command used for scheduling tasks to be executed sometime in the future.

```
OperatingSystem@Lab:~$ scheduling --syntax
```

    m h dom mon dow command
        m : minute   h : hour
        dom : day of month   mon : month
        dow : day of week
    * : means any step values

```
OperatingSystem@Lab:~$ cronjob
```

A cronjob is a Linux command used for scheduling tasks to be executed sometime in the future.

```
OperatingSystem@Lab:~$ scheduling --syntax

        m h dom mon dow command
                m : minute                h : hour
                dom : day of month        mon : month
                dow : day of week
        * : means any step values

OperatingSystem@Lab:~$ clear
```

```
OperatingSystem@Lab:~$ 1 0 1 * * ./CronJob.sh
```

That means run CronJob.sh on the first minute of the first day of each month

```
OperatingSystem@Lab:~$ clear
```

OperatingSystem@Lab:~$ BashScripter --ebook

For bash scripting you can read this ebook in github:

OperatingSystem@Lab:~$ BashScripter --ebook

For bash scripting you can read this ebook in github:

OperatingSystem@Lab:~$ clear

OperatingSystem@Lab:~$ BashScripter --cheatsheet

this is a cool and complete cheat-sheet for bash scripting:

OperatingSystem@Lab:~$ BashScripter --cheatsheet

this is a cool and complete cheat-sheet for bash scripting:

OperatingSystem@Lab:~$ clear

OperatingSystem@Lab:~$ shutdown 60

```
████████╗██╗  ██╗ █████╗ ███╗   ██╗██╗  ██╗    ██╗   ██╗ ██████╗ ██╗   ██╗██╗
╚══██╔══╝██║  ██║██╔══██╗████╗  ██║██║ ██╔╝    ╚██╗ ██╔╝██╔═══██╗██║   ██║██║
   ██║   ███████║███████║██╔██╗ ██║█████╔╝      ╚████╔╝ ██║   ██║██║   ██║██║
   ██║   ██╔══██║██╔══██║██║╚██╗██║██╔═██╗       ╚██╔╝  ██║   ██║██║   ██║╚═╝
   ██║   ██║  ██║██║  ██║██║ ╚████║██║  ██╗       ██║   ╚██████╔╝╚██████╔╝██╗
   ╚═╝   ╚═╝  ╚═╝╚═╝  ╚═╝╚═╝  ╚═══╝╚═╝  ╚═╝       ╚═╝    ╚═════╝  ╚═════╝ ╚═╝
```

(◍•‿•)