



UNIVERSIDADE
VILA VELHA
ESPIRITO SANTO

Terminal

Prof. Jean-Rémi Bourguet

Sistemas Operacionais

- **Concha/Casca:** camada **mais externa** ao redor do sistema operacional!



1900



1904



1909



1930



1948



1958



1961



1971



1998



1999

Shell

- ▶ Console: terminal físico que reconhece as teclas pressionadas
- ⇔ Terminal: ambiente de texto (input/output).
- ⇔ Interface de Linha de Comandos (CLI): meio de interagir com prog.
- ▶ Prompt de comando: indicar a prontidão para aceitar comandos.
- ▶ Shell: interpretador de linha de comando.



Pseudo terminais

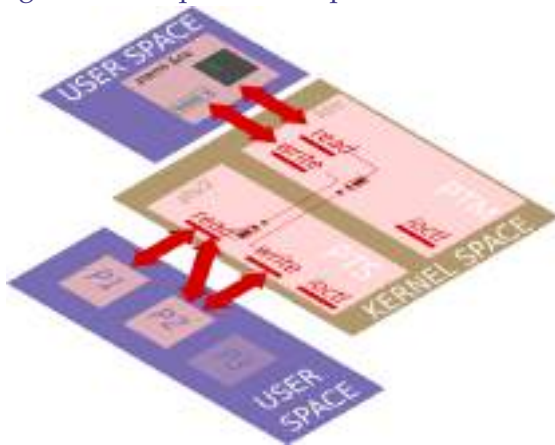
- ▶ Um pseudo-terminal é um **emulador de terminal**.
- ▶ Para abrir o pseudo terminal **CTRL + Alt + T**, para fechar **CTRL + D**.



Bastidores

- ▶ Cada **pseudo-terminal** ganha um arquivo de dispositivo no sistema.

```
$ tty  
/dev/pts/0
```



- ▶ **Escreva no arquivo de dispositivo** a partir de **outro pseudo-terminal**!

```
$ echo Hello World > /dev/pts/1
```

Teletypes

- ▶ **tty** significa **teletypewriter** que utilizava papel como tela.
- ▶ Os **tty** são tecnicamente conhecidos como **consoles virtuais**.



You Tube

 https://www.youtube.com/watch?v=AwqryPuw1_w

Operação multiusuário

Essa utilização pelos usuários poderá ser local ou remota:

- ▶ **Local:** quando eles estiverem **operando diretamente na máquina.**
- ▶ **Remota:** quando eles estiverem **operando via rede** (SSH ou Telnet).



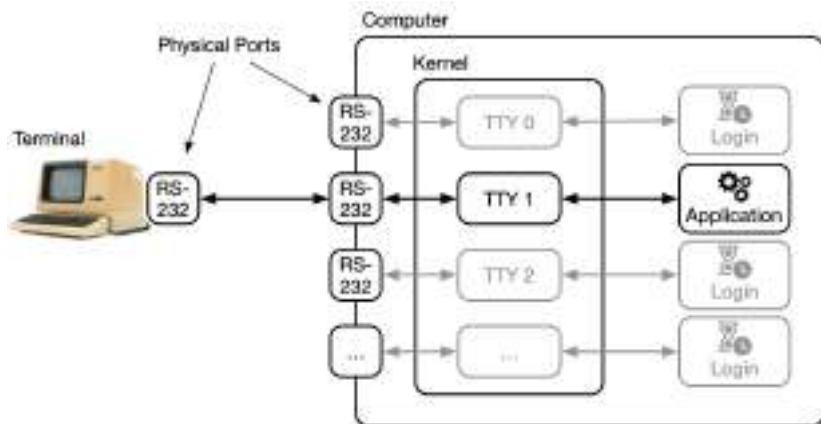
Acessando os terminais locais (tty)

- ▶ Existem **terminais** de **shell puro** (fora de ambientes gráficos).
- ▶ São do tipo **full screen** e servem pra **mostrar o sistema no monitor**.



Acessando os terminais locais (tty)

- ▶ **Começa com `tty1`** (terminal de log), `tty2` é um terminal gráfico!
- ▶ **`tty0`** não aparece: faz a **comunicação direta** com a **placa de vídeo**.



Acessando os terminais locais (tty)

- ▶ Alterne para um **outro terminal**, pressionando CTRL + Alt F3.
- ▶ **Faça o login** como sendo um **usuário** configurado na instalação!



Acessando os terminais locais (tty)

- ▶ No **prompt de login** do terminal primário digite **root** e **ENTER**.

```
Debian GNU/Linux squeeze mac tty1  
mac login: root  
Password:
```

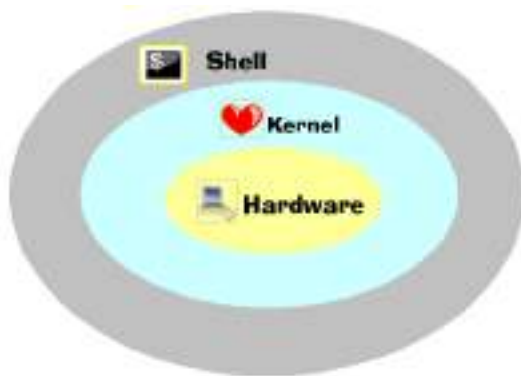
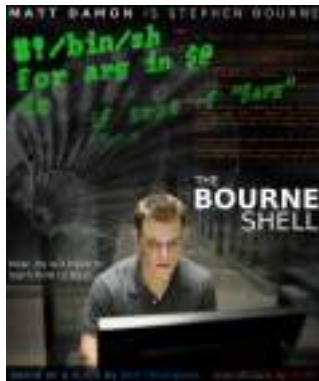
- ▶ Digite a **senha**, atribuída na instalação e **aparecerá um prompt**:

```
root@mac:~#
```



Shell

- ▶ **Shell** foi desenvolvido por **Stephen Bourne**, dos **laboratórios AT&T**.
- ▶ Bourne **shell**, ou **sh**, foi o **Shell padrão** do **UNIX Versão 7** (1977).



- **Stallman** e a **FSF** consideraram um **shell gratuito** executando scripts!



BASH

- ▶ **BASH** significa **B**ourne-**A**gain **S**hell (Born again:).
- ▶ Escrita por **Brian Fox** (FSF) em 1989 no **Projeto GNU** licenciado GPL.



- **Sexta linguagem** mais utilizada **no mundo!**



 <https://survey.stackoverflow.co/2023/#technology>

- **Previsão de tempo** de uma **cidade específica** com o **serviço wttr.in**:

user@mac:~\$ **nano weather.sh**



```
GNU nano 4.0 weather.sh
#!/bin/bash

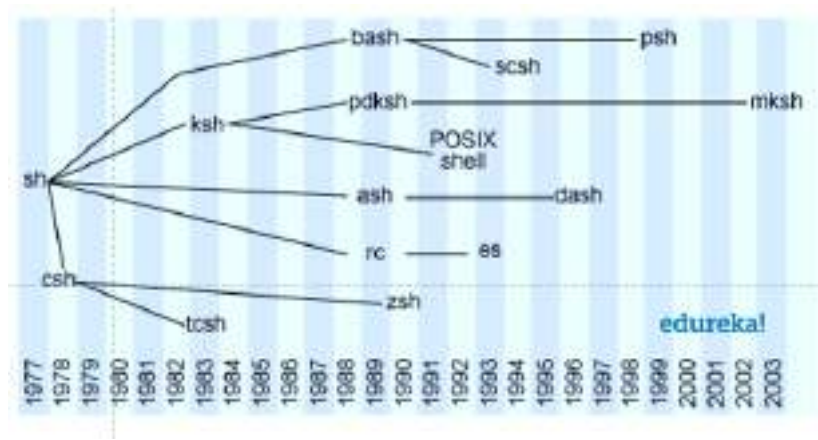
# Define the weather function
weather(){
curl -s "wttr.in/$1"
}

# Call the function with an argument
weather "$1"

^C Obter Ajuda  ^O Gravar      ^W Onde está?  ^R Recort txt
^X Salr        ^R Ler o arg  ^\ Substituir  ^U Colar txt
```

user@mac:~\$ **bash weather.sh "Vila_Velha"**

- ▶ A **sintaxe** do **comando Bash** é um **superconjunto** do **Bourne shell**.
- ▶ A **sintaxe inclui** ideias do **KornShell** (ksh) e do **C shell** (csh).



ZShell

- ▶ Por default, **Debian**, as filhas e a maioria das distros utiliza o **BASH**.
- ▶ Era o padrão em **macOS** antes de 2019, depois se tornou **ZSH** (e **WSL**).



https://www.youtube.com/watch?v=_DiEbmG3lU8

- Por que ohmyzsh? Você pode **customizar** com tema (e.g. agnoster).

root@mac:~# **apt install fonts-powerline**

user@mac:~\$ **nano ~/.zshrc**

```
CNS nano 4.0 /home/jean/.zshrc
# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load -- if set to "random", it will
# load a random theme each time Oh My Zsh is loaded. In which case,
# to know which specific one was loaded, run: echo RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="agnoster"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in $ZSH/themes/
# If set to an empty array, this variable will have no effect.
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

[ 104 linhas lidas ]
^G Obter Ajuda ^G Gravar ^M Onde está? ^N Recortar txt ^C Justificar ^O Pos atual
^X Salr ^X Ler o arg ^L Substituir ^U Colar txt ^V Verificar ^P Ir p/ linha
```

<https://github.com/powerline/fonts>

ZShell

- Teste **um TAB** depois do **ch**, um **segundo TAB** e um **TAB depois do -**.

```
user@mac:~$ chsh -s /bin/bash
```

```
user@mac:~$ zsh
```

```
user@mac:~$ bash
```



* Para efetivar a troca para o usuário: sair da sessão (Ctrl Alt Del)

Atalhos

- ▶ OS possui **muitos atalhos** para **facilitar a utilização** dos terminais.



- ▶ **Ctrl+Alt+T**: Abrir T. **Ctrl+D**: Log off.
- ▶ **Ctrl+C/** Matar. **Ctrl+L**: Limpar tela.
- ▶ **Ctrl+U**: Limpar linha ← (**Ctrl+K**: →).
- ▶ **Ctrl+E**: Ir fim de linha → (**Ctrl+A**: ←).
- ▶ **Ctrl+Y**: Colar. **Ctrl+_**: Desfazer
- ▶ **Ctrl+R** (**Ctrl+G**): Buscar (Cancel).





Root vs. Usuários

O prompt do sistema poderá terminar com um:

- ▶ **caractere cerquilha (#)** que só é utilizada com o root:

```
# whoami  
root
```

- ▶ **cifrão (\$)** que é utilizada com qualquer outro usuário:

```
$ whoami  
linus
```



Estrutura de diretórios

- **Estrutura básica de diretórios** ligada à **raiz da árvore** de diretórios (/).



Cada diretório

`/bin`

- Os arquivos **essenciais** durante a **recuperação em panes** no sistema.



* O diretório `/bin` é um link simbólico (atalho) para o diretório `/usr/bin`

Cada diretório

/boot

- ▶ Contém os arquivos **do bootloader GRUB** que inicializa o Kernel
- ▶ Contém **imagens do Kernel** que irá **montar FS** e **iniciar processos**.



Cada diretório

`/dev`

- ▶ Contém **arquivos de ligações** com **dispositivos de hardware** devices.
- ▶ Permitem justamente **conectar o sistema operacional ao hardware**.



Cada diretório

/etc

- Contém a grande maioria dos **arquivos de configuração** do OS e rede.



"This slide needs work."

Cada diretório

`/home/<user>`

- ▶ Contém os **arquivos, documentos e configurações dos usuários**.
- ▶ Cada usuário possui um **subdiretório próprio** dentro do home.



Cada diretório

/lib

- ▶ Contém as **bibliotecas, módulos do kernel e drivers de dispositivos**.
- ▶ **Bibliotecas utilizadas** dos **executáveis existentes** em **/bin** e **/sbin**.



`/lost+found`

- ▶ Files **encontrados** que tenham **perdido o vínculo com os inodes**.
- ▶ **Corrupção: recuperação de arquivos** após **falhas de energia** (fsck).



Cada diretório

/media

- ▶ Trata-se de **ponto de montagem automática** de **mídias removíveis**.
- ▶ Exemplos são **disquete**, **CD-ROM**, **DVD**, **pendrive**, etc.



/mnt

- **Ponto de montagem manual** de **FS** para administradores de sistema.



Cada diretório

/opt

- ▶ Contém **programas** que **não fazem parte da distribuição** (Chrome).
- ▶ Instalação e desinstalação, e **não gerenciados** pelo sistema de pacotes.



Cada diretório

`/proc`

- ▶ É um **diretório virtual**. Na verdade, é um **filesystem virtual**.
- ▶ Gerado **constantemente pelo kernel** (e.g. **ocupação de memória**).



Cada diretório

/root

- ▶ Diretórios em /home podem não estar dispo em **situações de crise**.
- ▶ Neste caso, apenas a **partição raiz do sistema** será montada.



Cada diretório

`/sbin`

- ▶ **Similar ao `/bin`: administração e manutenção em `panes` (system).**

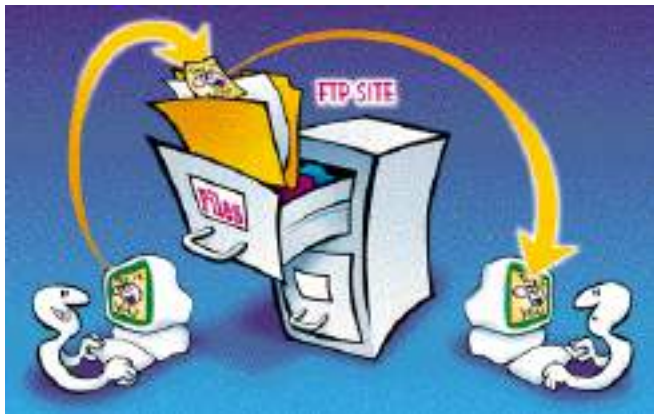


* O diretório `/sbin` é um link simbólico (atalho) para o diretório `/usr/sbin`

Cada diretório

`/srv`

- ▶ Dados e arquivos relacionados a **serviços fornecidos** pelo sistema.
- ▶ Usados pelos **servidores web**, **FTP**, ou outro **serviço de rede**.



`/sys`

- **sys** filesystem: interface para interagir com o **kernel** e o **hardware**.



Cada diretório

/tmp

- Utilizado para **gravar dados temporários** de programas, sessão.
- Por norma, seu conteúdo será **totalmente apagado a cada boot**.



Cada diretório

`/usr` (**u**nix **s**hared **r**esources)

- ▶ Contém a **maior parte dos arquivos** que **possam ser compartilhados**.
- ▶ Diretórios `usr/bin`, `usr/sbin` ou `usr/lib`.



Cada diretório

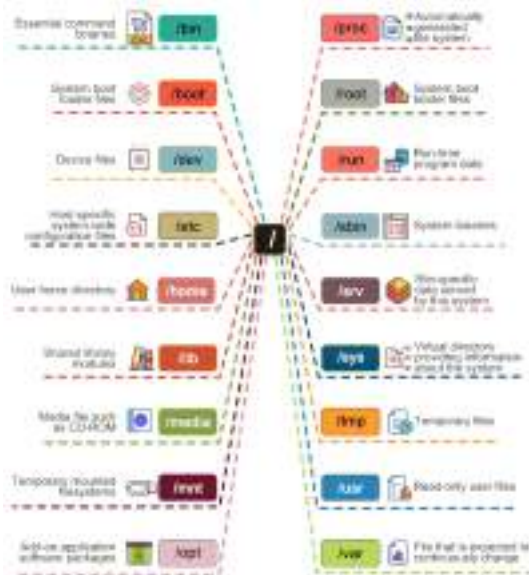
`/var`

- Contém **arquivos** com o **conteúdo variável**: **logs**, **pool**, **dados** de BD.



Filesystem Hierarchy Standard

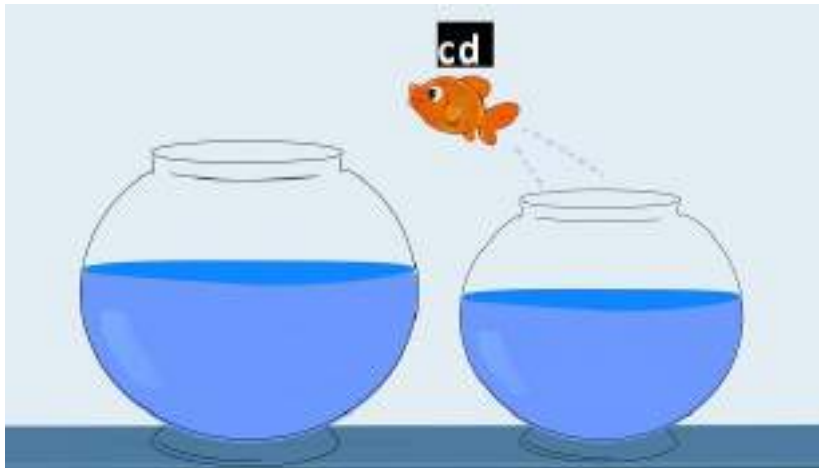
FHS.gif



cd (change directory)

- **Navegar em diretórios: sair de um diretório e entrar em outro.**

\$ cd diretório



cd (change directory)

- ▶ Vamos se posicionar em `/usr`. Se digitarmos:

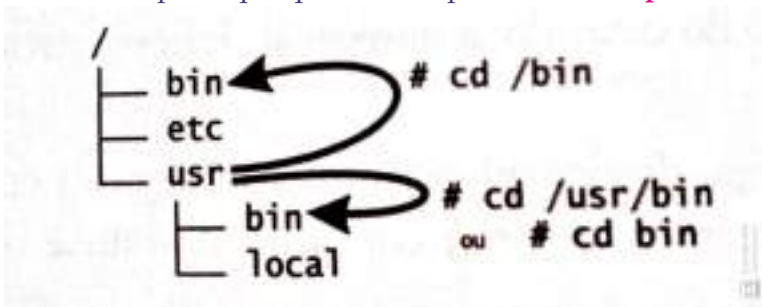
`cd /bin`

Já sabemos que iremos diretamente para o diretório `/bin` (linha 2).

- ▶ No entanto, se **omitirmos a barra**, digitando:

`cd bin`

- ▶ O sistema interpreta que queremos ir para um `bin` **a partir do local**.



Subindo e descendo

- ▶ Quando tratamos de **navegação** trata-se de **subir e descer** diretórios
- ▶ Se estivermos em `/usr` e digitarmos: `# cd ..` subiremos até raiz (`/`).



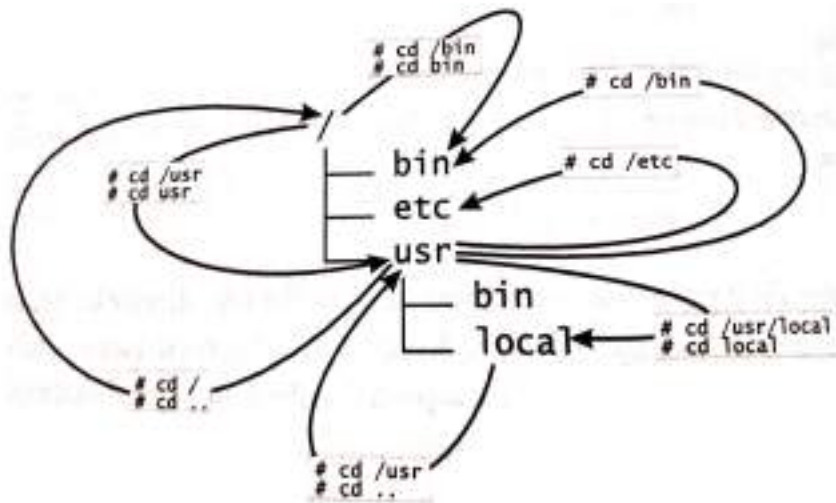
```
cd ../..../..
```



```
cd ..  
cd ..  
cd ..  
cd ..
```

Subindo e descendo

- ▶ Há **mais de uma possibilidade** para **executar a mesma navegação!**



Subindo e descendo

- O que acontecerá se for **tentar subir** a partir **da raiz**?

```
root@mac:/# cd /
```

```
root@mac:/#
```



cd (change directory)



Comando	Resultado
# cd /usr	Vai para o diretório /usr
# cd /usr/bin	Vai para o diretório /usr/bin
# cd bin	Vai para o diretório bin a partir do diretório atual.
# cd ..	Sobe um nível de diretório.
# cd ../etc	Sobe um nível de diretório e entrar em etc.
# cd	Vai para o diretório home do usuário.
# cd -	Volta para o diretório no qual estava antes (OLDPWD).



su (substitute user)

- ▶ Para **alternar para o usuário** "eriberto": # **su** **linus**
- ▶ Para **alternar para o root**, carregando o ambiente dele: # **su -**



su (substitute user)

- ▶ Informar a **senha do novo** a não ser que o **login de origem** seja o **root**.
- ▶ O **ambiente (environment)** são as variáveis de ambiente do usuário.

Chave	Função
-	carrega o ambiente do usuário que está sendo chamado.
-c COMMAND	comando é executado usando os privilégios do usuário.
-h	define o ambiente shell a ser usado com o usuário.

+ Para alternar para o usuário "linus", carregando o seu ambiente:

```
# su - linus
```

+ A mesma coisa mas mantendo o ambiente do usuário atual:

```
# su -c "apt install sl" root
```

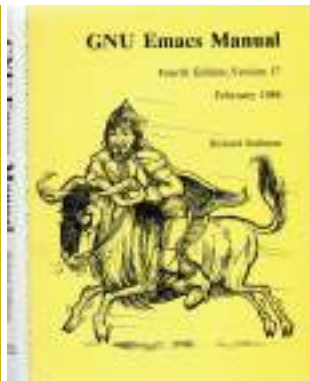
Edição de textos no shell

- ▶ Existem **vários editores de texto** para GNU/Linux **em modo Shell**.
- ▶ **vi e nano já vêm instalado** na maioria, se não em todas as distros.





- ▶ O **Emacs** é um editor de texto, usado por **programadores avançados**.
- ▶ A **primeira versão** foi **escrita em 1976** por **Richard Stallman**.



- **M** representa <Alt>, **C** representa <Control> e **S** representa <Shift>.



- Um **hífen** significa que **todos devem estar ativos** ao mesmo tempo.
- **Sequências separadas por espaço** denotam **execução sucessiva**.

Atalhos emacs



PRESS X THEY SAY...

- ▶ **C-x C-f**: Abrir um arquivo (**f**ile)
- ▶ **C-x C-s**: Salvar arquivo (**s**ave)
- ▶ **C-x C-w**: Salvar como (**w**hatever)
- ▶ **C-x C-c**: Fechar (**c**lose)

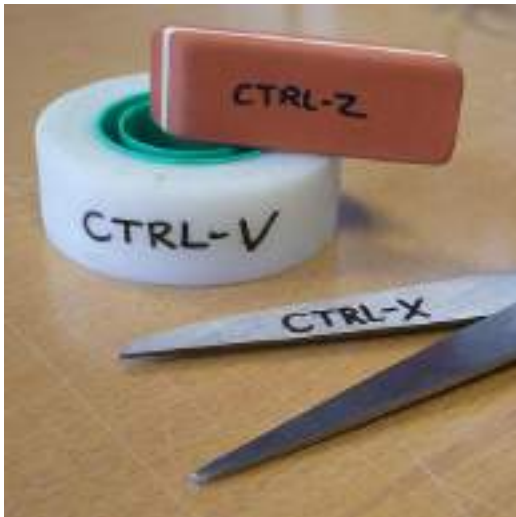


Atalhos emacs

- | | |
|--|---|
| → C-f : Avançar +1c (f oreward) | ← C-b : Voltar -1c (b ackward) |
| ↷ M-f : Avançar +1w (f oreward) | ↷ M-b : Voltar -1w (b ackward) |
| → C-e : Avançar +1l (e nd) | ← C-a : Voltar -1l (a bc) |
| ↓ C-n : Descer -1l (dow n) | ↑ C-p : Subir linha +1l (u p) |
| ⇓ M-v : Subir página +1p (...) | ⇓ C-v : Descer página -1p (...) |
| ↘ M-> : Ir no final da janela | ↘ M-< : Ir no início da janela. |



Atalhos emacs



- ▶ **C-SPACE**: Marcar
- ▶ **M-w**: Copiar
- ▶ **C-y**: Colar
- ▶ **C-w**: Recortar
- ▶ **C-_**: Desfazer
- ▶ **C-x h**: Selecionar tudo



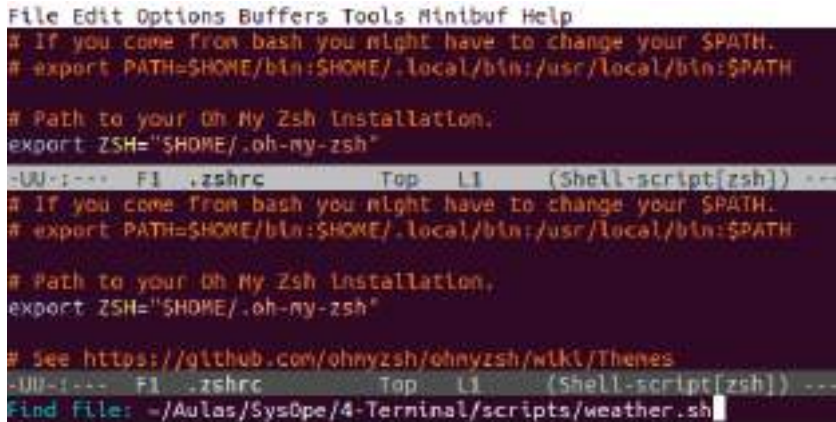
Atalhos emacs



- **C-x 2**: Split horizontal
- **C-x 3**: Split vertical
- **C-x o**: Leva o cursor
- **C-x 0**: Eliminar janela
- **C-x 1**: Voltar em uma tela
- **M-PDown** / **M-PUp**: Na outra
- **C-s**: Buscar para frente
- **C-r**: Buscar para tras

Atividade emacs

- ▶ Vamos abrir o arquivo de configuração: `$ emacs -nw ~/.zshrc`
- ▶ Abrir um nova tela **C-x 2** e carrega **C-x C-f** /path/to/weather.sh



```
File Edit Options Buffers Tools Minibuf Help
# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# Path to your Oh My Zsh Installation.
export ZSH="$HOME/.oh-my-zsh"

-UU--1-- F1 .zshrc Top L1 (Shell-script[zsh]) --
# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# Path to your Oh My Zsh Installation.
export ZSH="$HOME/.oh-my-zsh"

# see https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
-UU--1-- F1 .zshrc Top L1 (Shell-script[zsh]) --
Find File: ~/Aulas/SysOpe/4-Terninal/scripts/weather.sh
```

Atividade emacs

- ▶ Se deslocar até C-n **C-SPACE** C-n a função, marcar até a decla.
- ▶ Copiar **M-w**

```
File Edit Options Buffers Tools Sh-Script Help
```

```
#!/bin/bash
```

```
# Define the weather function
```

```
weather(){
```

```
curl -s "wttr.in/$1?m1"
```

```
}
```

```
# Call the function with an argument
```

```
weather "$1"
```

```
-UU-|- F1 weather.sh All L10 (Shell-script[bash]) --
```

```
# If you come from bash you might have to change your $PATH.
```

```
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH
```

```
# Path to your Oh My Zsh installation.
```

```
export ZSH="$HOME/.oh-my-zsh"
```

```
-UU-|- F1 .zshrc Top L1 (Shell-script[zsh]) ---
```

Atividade emacs

- ▶ Mudar de Tela **C-x o**, Ir no final do .zshrc **C-v** ou **C-n**
- ▶ Colar **C-y** e trocar o argumento por "Vila_Velha".

File Edit Options Buffers Tools Sh-Script Help

```
#!/bin/bash
```

```
# Define the weather Function
```

```
weather(){
```

```
curl -s "wttr.in/$1?m1"
```

```
}
```

```
# Call the function with an argument
```

```
weather "$1"
```

```
--UU<!-- F1 weather.sh All 110 (Shell-script[bash]) -->
```

```
# alias ohmyzsh="nate ~/.oh-my-zsh"
```

```
weather(){
```

```
curl -s "wttr.in/$1?m1"
```

```
}
```

```
# Call the function with an argument
```

```
weather "Vila_Velha"
```

```
--UU<!-- F1 .zshrc Top 11 (Shell-script[zsh]) -->
```

Atividade emacs

- ▶ Salvar **C-x C-s** e sair **C-x C-c**.
- ▶ Chamar zsh no terminal.



Comandos estendidos



- ▶ **M-x save-buffer**: Salvar
- ...
- ▶ **M-x search-**: Buscar
- ▶ **M-x search-forward**: Enter:
- ▶ **M-x search-forward-regexp**:
- ▶ **M-x cd**: Enter: change dir
- ▶ **M-x shell**: Enter: tornar shell

Doom emacs

- ▶ É uma **distribuição** ou **configuração aprimorada** do emacs (2017).

