

UNIX E BASH

cd muda de diretório (change directory)

cd / muda para o diretório root/raiz

ls - lista tudo que têm no diretório/pastas

ls -a lista tudo mas arquivo oculto (ficam com um . na frente)

ls * mostra tudo dentro do diretório, consegue ver todos os arquivos de todos os diretórios dentro

ls ? _____ as vezes vc não lembra de uma letra

```
→ ls ?rybe
arquivo1.txt
```

mkdir (make directory) cria um diretório

cd ~ é o home

variações do cd

- **cd .** referência o próprio diretório
- **cd ..** volta no diretório pai

pwd mostra o diretório completo (endereço) (print working directory)

cp (copy) nome do arquivo e o nome da cópia (você dá o nome)

cat abre o arquivo e mostra o conteúdo, pra sair **cd**.

mv (move) mover um arquivo (mv arquivo e o endereço pra onde vc quer que vá) ou muda o nome

rm (remove) nome do arquivo

rmdir (remove directory) nome do diretório

rm -rf remove o diretório mesmo se tiver arquivos ou outras pastas (remove forced)

less exibir parte de conteúdo por página para ir pra próxima página é o espaço e o Q volta

- dentro do less vc pode colocar / _____ e no sublinhado vc coloca a palavra chave
- **n** para passar pro próximo (next)

head mostra as 10 primeiras linhas

head -5 os cinco primeiros valores

tail mostra as 10 últimas linhas

grep pesquisa fora do arquivo, porém precisa direcionar o mesmo, têm que colocar o nome certo na pesquisa

```
→ grep Toy filmes.txt
Toy Story 3
Toy Story 2
Toy Story 1
```

- **grep -i** ' ' pesquisa menos sensível (ignore)
- **grep -iv** ' ' vai trazer todos os valores que não corresponde com a busca (inverse)
- **grep -in** ' ' vai dar a linha que o mesmo se encontra (number)
- **grep -ic** ' ' quantidade de vezes que aparece (count)

wc (word count) quantidade de letras, aparece 3 valores (quantidade de linhas, quantidade de palavras e quantidade de caracteres)

- **wc -w** quantidade de palavras
- **wc -l** quantidade de linhas
- **wc -c** quantidade de caracteres

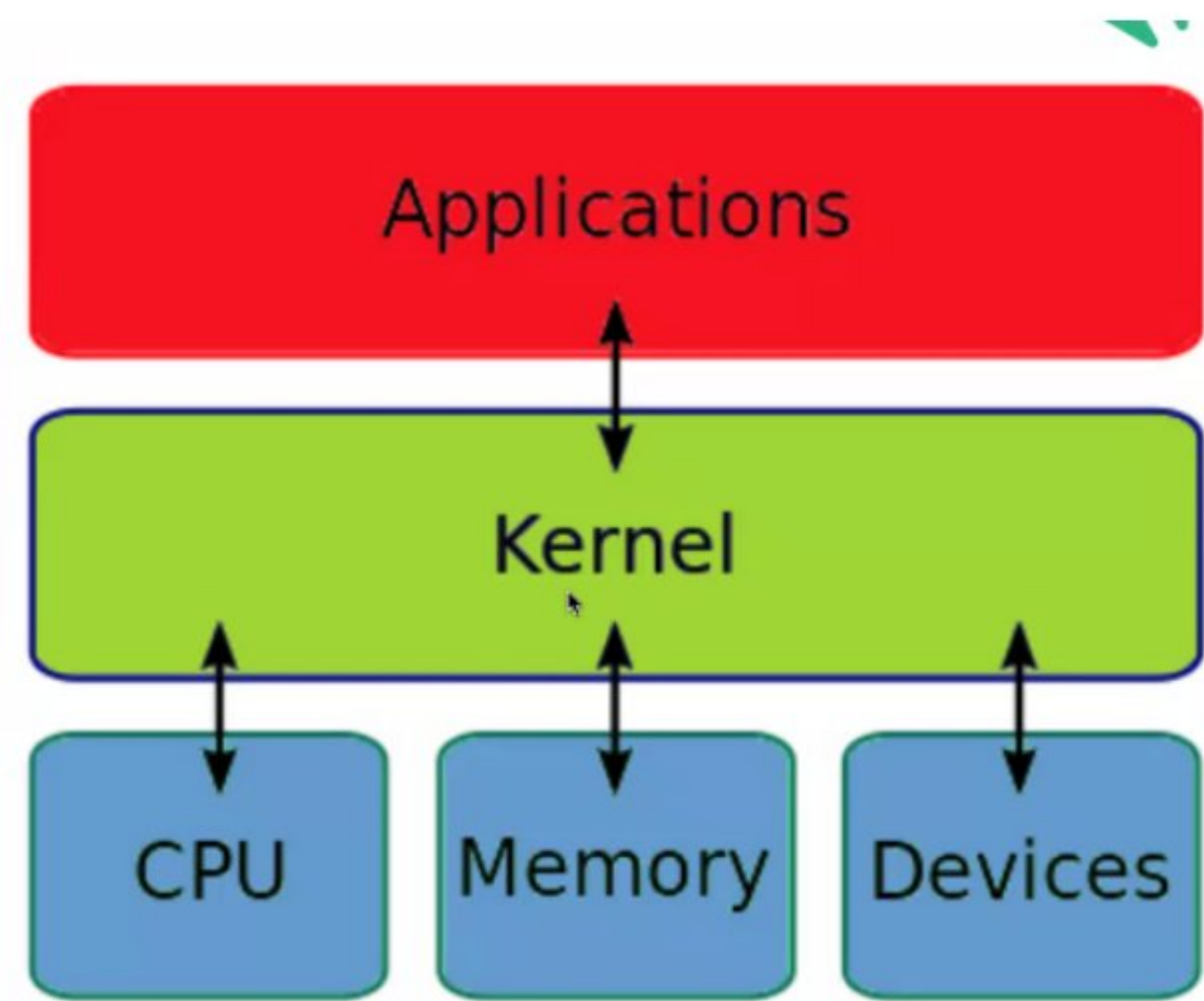
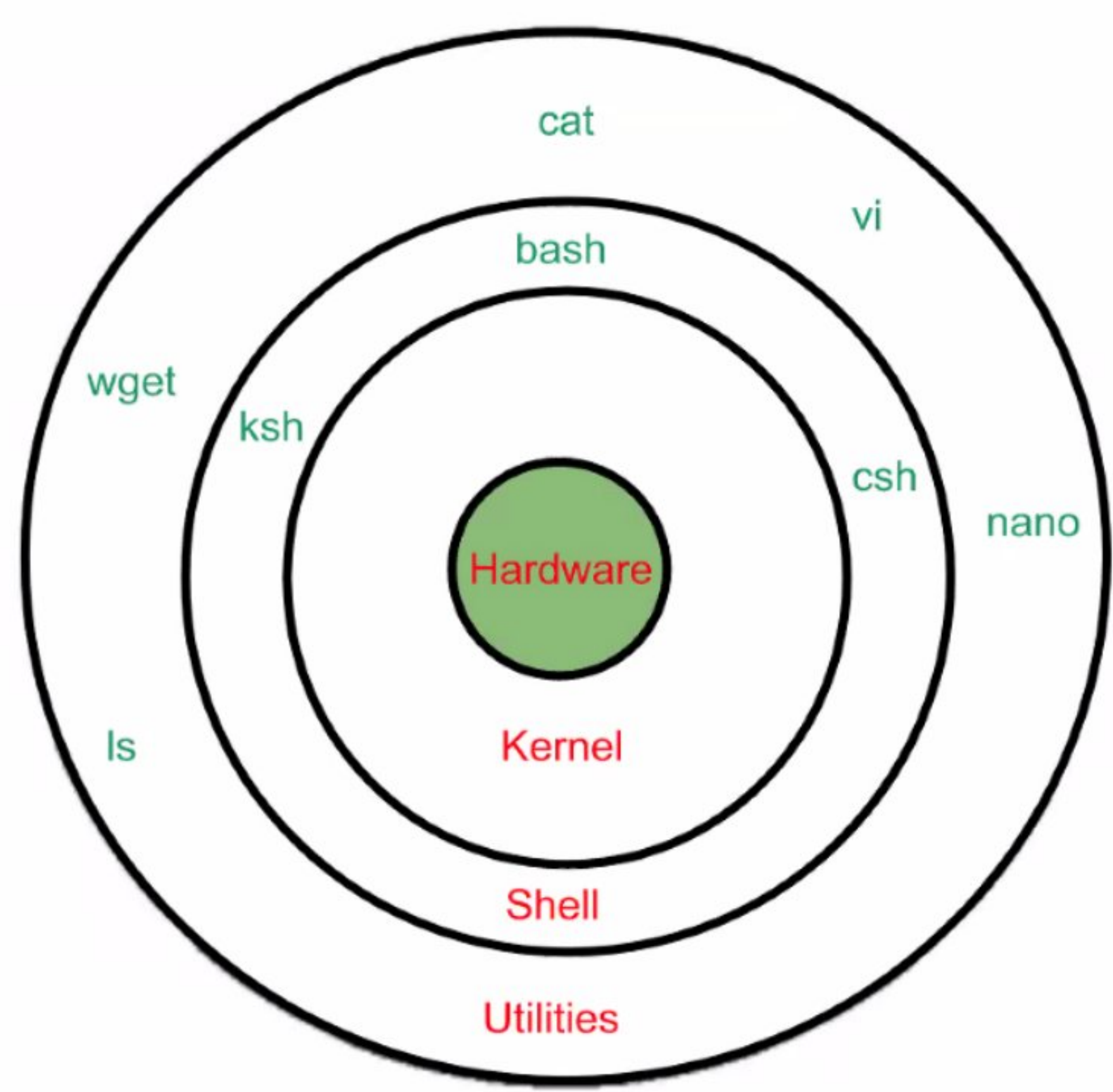
touch cria um arquivo não precisa de comando, só digitar sem espaço

man (manual) qualquer comando na frente, vai acessar o manual daquele comando

whatis o comando na frente, vai mostrar o que o comando faz

apropos comando que eu quero, e vai mostrar todos os comandos que fazem isso
vim abre edição de arquivo, pra sair e salvar :wq

Comando	Significado
ls	lista arquivos e diretórios
ls -a	lista todos os arquivos e diretórios
mkdir	cria um diretório
cd <i>diretório</i>	muda para o diretório nomeado
cd	muda para o diretório inicial
cd ~	muda para o diretório inicial
cd ..	muda para o diretório pai
pwd	mostra o caminho do diretório atual
cp <i>arquivo1 arquivo2</i>	copia arquivo1 e o chama de arquivo2
mv <i>arquivo1 arquivo2</i>	move ou renomeia arquivo1 para arquivo2
rm <i>arquivo</i>	remove um arquivo
rmdir <i>diretório</i>	remove um diretório
cat <i>arquivo</i>	exibe um arquivo
less <i>arquivo</i>	exibe um arquivo, uma página por vez
head <i>arquivo</i>	exibe as primeiras linhas de um arquivo
tail <i>arquivo</i>	exibe as últimas linhas de um arquivo
grep ' <i>palavras-chave</i> ' <i>arquivo</i>	procura por palavras-chave no arquivo
wc <i>arquivo</i>	conta o número de linhas / palavras / caracteres no arquivo
*	corresponde a qualquer número de caracteres
?	corresponde a um caracter
man <i>comando</i>	lê a página do manual online do comando informado
whatis <i>comando</i>	breve descrição do comando informado
apropos ' <i>palavra-chave</i> '	corresponde a comandos com palavras-chave em suas páginas de manual



```
unix_tests mv backup/trybe_backup.txt backup2
```

este arquivo estava dentro de outro diretório, pegamos ele e mandamos para outro diretório sem sair

UNIX E BASH II

cat ctrl d sai do cat
cat > vamos atribuir e substituir valores para um determinado arquivo criado imediatamente
exemplo:

```
→ cat > cores.txt
amarelo
vermelho
azul
```

valores adicionados

cat >> adicionar e editar mais informações ao meu arquivo

```
→ cat >> cores.txt
marrom
preto
laranja
```

cat arquivo1 arquivo2 **>** arquivo3 junta todas as informações de outros 2 arquivos em um outro arquivo
sort **<** ordena arquivos somente para mim
sort **<** arquivo1 **>** arquivo4 envia os valores para um arquivo em forma ordenada
who mostrar qm está logando
pipe **|** jogar um comando no outro

exemplo:
who | sort

```
→ who | sort
gustavo.caetano console Sep 25 09:23
gustavo.caetano ttys000 Sep 25 09:55
```

who | wc -l

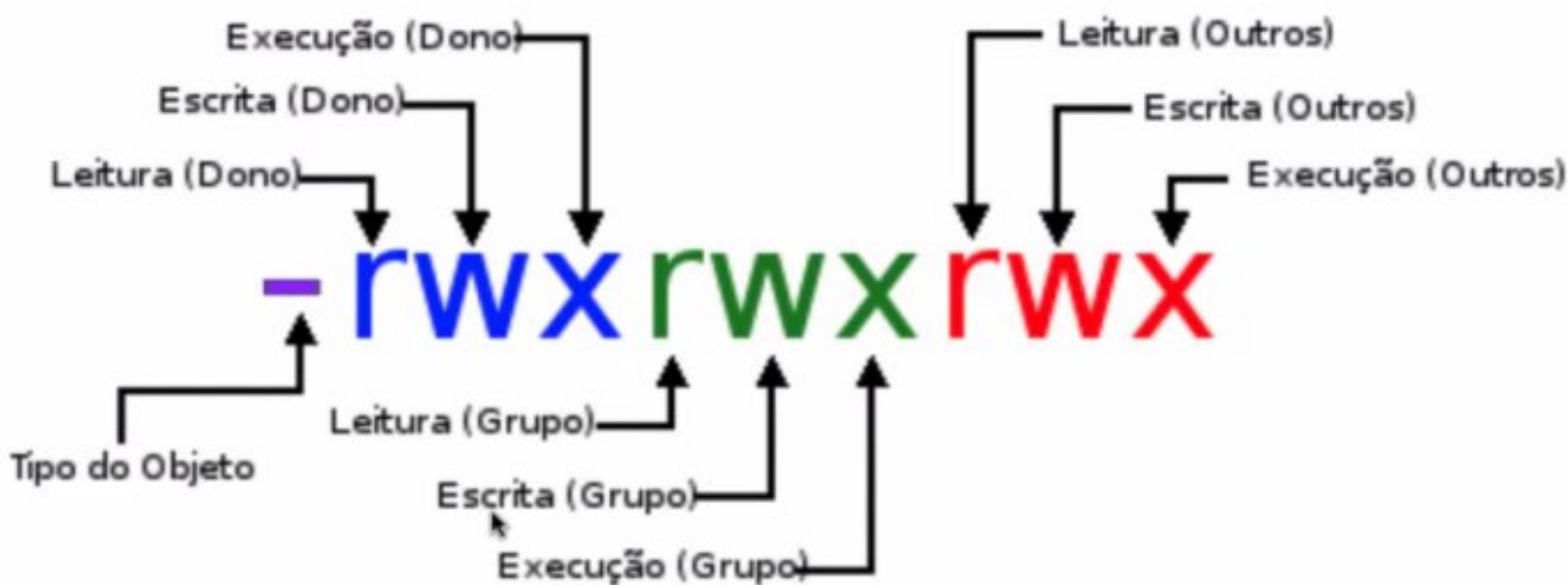
```
→ who | wc -l
2
```

ls -l mostra outras informações que o ls sozinho não mostra (acessos de qm pode editar, usuário, o grupo pertencente, o dia o mês, e à hora da última alteração).

Permissões	Links	Proprietário	Grupo	Tamanho	Data e Hora	Nome
drwxr-xr-x	2	root	root	4096	Out 19 09:10	composer/

- r** (read) leitura permitida do arquivo ou diretório
- w** (write) pode editar, alterar o valor do diretório
- x** (execute) permite executar um arquivo ou acessar um diretório pelo comando cd
- no início da sequência indica um arquivo comum
- d** no início da sequência indica um diretório

Dono => **u**
Grupo => **g**
Outros => **o**



chmod (change modification)

- **chmod u** modificar só pra usuário
- **chmod g** modificar para o grupo
- **chmod o** modificar para outros usuários
- **chmod a** para todo mundo
- **chmod _ -** tirar permissão
- **chmod _ +** adicionar permissão
- **chmod _ =** sem modificar nada

ps (process) lista o que está sendo executado no momento, podem estar em execução, em background ou suspensos

ps -A mostra tudo do computador

sleep para o terminal por um determinado tempo exemplo

- **sleep 5** (5 de 5 segundos)
- **sleep __ &** abre um processo e deixar em background (__ é o tempo)

bg faz o último processo executado ir para o background

jobs lista que exibe todos os processos executados e diz seus status

fg (for grounded) faz o processo voltar à ser executado

- **fg %_** listar qual dos processos que vc quer

kill %_ encerra processo que vc seleciona

- **kill ____** (número do processo) tbm encerra
- têm processos que não aceitam ser finalizados
- **kill -9 ____** (número do processo) força à encerrar processos que não aceitavam ser finalizados

find localiza nos ser nome, data, tamanho e tipo

- **find . -name “__”** todos os arquivos que terminam na extensão citada (“.txt”)
- **find . -type d** todos os diretórios
- **find . -type f** todos os arquivos
- **find ./teste -name “exemplo*”** tanto arquivos como diretórios comecem com um trecho
- **find ./teste -type f -name “exemplo*”** arquivos que comecem por algum trecho
- **find ./teste -type d -name “exemplo*”** diretórios que comecem por algum trecho

history comando que mostra o histórico de comando que você usou no terminal, os tamanhos do quanto ele vai mostrar pode ser configurado

- **history** comando geral
- **history | tail** pegar os últimos 10 comandos
- **history | head** pegar os 10 primeiros

echo utilizado para exibir mensagem na tela ou em um arquivo

echo “_____” (mensagem dentro dos aspas)

echo “_____” > arquivo (coloca o texto dentro de arquivos)

ctrl z deixa o processo suspenso

ctrl c termina processo

ctrl d fecha o cat

```
alex@DESKTOP-V0SE7UF:~/unix_tests$ sort < skills2.txt | head -3 > top_skills.txt
```



```
kaka@DESKTOP-NNN50RV:~/unix_tests$ chmod a+rw bunch_of_things.txt
kaka@DESKTOP-NNN50RV:~/unix_tests$ ls -l
total 24
-rw-rw-rw- 1 kaka kaka 38 Sep  8 19:08 bunch_of_things.txt
```

cat vs echo? diferença?

à diferença é que o cat têm certos tipos de estruturas, ele abre o arquivo para editar, diferente do echo, que abre uma saída para inserir um arquivo

Comando	Significado
comando > arquivo	redireciona a saída padrão para um arquivo
comando >> arquivo	anexa saída padrão a um arquivo
comando < arquivo	redireciona a entrada padrão de um arquivo
comando1 comando2	canaliza a saída do comando1 para a entrada do comando2
cat arquivo1 arquivo2 > arquivo0	concatena arquivo1 e arquivo2 em arquivo0
sort	ordena dados
who	lista os usuários atualmente logados
ls -lag	lista direitos de acesso para todos os arquivos
chmod [options] arquivo	altera os direitos de acesso para o arquivo nomeado
comando &	executa o comando em segundo plano
^C	elimina o trabalho em execução em primeiro plano
^Z	suspende o trabalho em execução em primeiro plano
bg	retorna processos que estão suspensos
jobs	lista processos suspensos e em segundo plano
fg %1	reinicia o primeiro processo suspenso
kill %1	encerra o primeiro processo suspenso
ps	lista processos atuais
kill 26152	encerra o processo de número 26152