
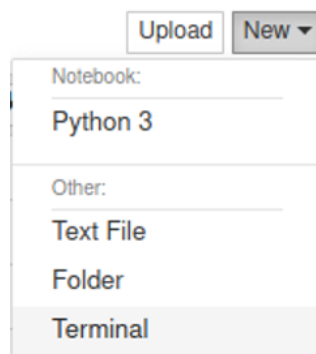


1 – Introdução

Essa apostila é destinada àqueles com pouco ou nenhum conhecimento em linha de comando Linux. Os comandos aqui abordados são: `ssh`, `pwd`, `cd`, `ls`, `cp`, `mv`, `touch`, `rm`, `less`, `mkdir`, `ln`, `wc`, `grep` e `nohup`. Se você já está familiarizado com eles, a leitura é recomendável para recordar. Do contrário, a leitura é obrigatória, pois esses comandos serão bastante utilizados durante o curso.

Para quem não está acostumado com linha de comando, procuramos inserir no texto comparações de operações similares no Windows (tela gráfica). Além disso, é bom saber previamente que um diretório nada mais é que uma pasta do Windows. Quando o texto se refere a criar um diretório, a operação é idêntica a criar uma pasta. Também vale dizer que a estrutura de diretórios é um pouco diferente: o C: do Windows é o correspondente ao / (barra/raiz) no Linux.

Além da leitura, sugerimos um treinamento prático com linha de comando em Linux. Para praticar os comandos desta apostila você deve clicar no botão . Após o sistema ter sido carregado você deve clicar em New -> Terminal.



2 – “Atalhos” do sistema

Antes de entrar na parte dos comandos, é interessante saber sobre os *hard links* do sistema. Os *hard links* nada mais são do que atalhos que facilitam a vida de quem usa linha de comando constantemente. Segue a lista de alguns bastante usados:

- 1) Atalho para o diretório corrente: O caminho completo do diretório corrente pode ser substituído por “.” (ponto). Quando usarmos “.”, o Linux vai interpretar que queremos fazer algo no diretório ao qual nos encontramos.
- 2) Atalho para o diretório pai do diretório corrente: O caminho completo do diretório pai pode ser substituído por “..” (ponto ponto). Quando usarmos “..”, o Linux vai interpretar que queremos fazer algo no diretório pai do diretório corrente, isto é, no diretório exatamente acima do diretório atual.

3 – ssh

Além de usar o Linux no seu computador local, você pode querer acessar um servidor onde armazena seus dados e executa processos. Se for o caso, você deve fazer isso usando o comando ssh, que é um programa para login remoto. Para usar o ssh você deve passar algumas informações sobre o servidor ao qual você deseja acessar, como:

- 1) Porta ssh utilizada pelo servidor remoto: o padrão ssh é a porta 22, mas isso depende do administrador da rede. É definida pelo parâmetro -p.
- 2) Login: assim, como você tem um nome de usuário no Windows ou no Linux do seu desktop, você precisa especificar qual é o seu usuário no seu servidor remoto.
- 3) Endereço de rede ou IP: assim como o correio só consegue lhe entregar uma carta com seu endereço e CEP, o servidor que você deseja acessar tem um endereço de rede.

A sintaxe padrão é “ssh -p porta_do_servidor login@endereco_de_rede”. Após o comando ser executado, será solicitada sua senha de acesso, como mostrado abaixo:

```
[osvaldo@desktop ~]$ ssh -p 2323 osvaldo@173.168.23.12
osvaldo@173.168.23.12's password:
```

Caso a senha inserida esteja correta, acessamos o servidor ou, popularmente, logamos no servidor. Ao logar, o sistema me redireciona para o meu homedir. O homedir é um diretório que só eu tenho acesso, ou seja, tudo que for gravado aqui não ficará disponível para os outros usuários.

Para este comando não temos como treinar na prática, mas lembre-se que quando for necessário fazer uma conexão remota por linha de comando você pode recorrer a este comando.

Mas qual é a localização do meu homedir no sistema?

4 - pwd

O comando pwd (print working directory) exibe o caminho completo (desde a raiz) do diretório corrente.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d8kwzsurre:~$ pwd
/home/jovyan
```

Ao usar esse comando no meu homedir, descobri que o caminho completo dele no sistema é: “/home/jovyan”.

5 – cd

O comando `cd` (change directory) serve para você modificar o diretório do sistema na qual você se encontra. É similar a ir para outra pasta qualquer no Windows, ou popularmente entrar em uma pasta. No nosso caso, vamos entrar no diretório `curso`, que é um subdiretório do meu homedir.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dimsie94j:~$ cd /home/jovyan/curso/
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dimsie94j:~/curso$
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dimsie94j:~/curso$ pwd
/home/jovyan/curso
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dimsie94j:~/curso$
```

Dica: Ao estudar, tente usar os comandos que você já aprendeu anteriormente. Isso vai facilitar bastante o seu aprendizado. No exemplo anterior usamos o comando `pwd` após a operação de mudança de diretório para verificar se o comando foi bem sucedido.

Como faço para visualizar o conteúdo do subdiretório `curso`?

6 – ls

O comando `ls` (List Directory) lista todo o conteúdo do diretório corrente, incluindo arquivos e subdiretórios.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls
exomas  genoma  Homo_sapiens.GRCh38.dna.primary_assembly.fa  mae.R1.fq  mae.R2.fq
pai.R1.fq  pai.R2.fq
```

Usando o comando `ls`, descobri que dentro do diretório corrente existem dois subdiretórios (`exomas` e `genoma`) e mais cinco arquivos. Os subdiretórios são exibidos com cor diferente que depende das configurações do seu sistema Linux ou do programa `putty` (no meu caso, em azul).

O comando `ls` pode ser usado com uma ou mais opções. Se quisermos obter detalhes dos subdiretórios e arquivos usamos a opção `-l` (long listing). Ao usar essa opção, podemos visualizar várias informações do arquivo que foram omitidas no formato

curto, como permissões (1ª coluna), usuário proprietário (3ª coluna), tamanho (5ª coluna), data e hora da criação (6ª, 7ª e 8ª coluna). O arquivo “Homo_sapiens.GRCh38.dna.fa”, por exemplo, tem 11021 bytes.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls -l
total 36
drwxr-xr-x 2 jovyan jovyan 4096 Apr 28 15:10 exomas
drwxr-xr-x 2 jovyan jovyan 4096 Apr 28 15:10 genoma
-rw-r--r-- 1 jovyan jovyan 11021 Apr 28 15:10 Homo_sapiens.GRCh38.dna.fa
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R1.fq
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R2.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R1.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R2.fq
```

No Windows, os detalhes dos arquivos e subdiretórios podem ser obtidos utilizando o botão direito do mouse, clicando em “Exibir”, depois em “Detalhes” no diretório corrente. Esse comando vai gerar uma lista que contém somente informações gerais. Detalhes mais específicos são obtidos clicando com o botão direito do mouse em cima de cada arquivo, e depois em “Propriedades”.

Outra opção do comando `ls` é mostrar os arquivos e diretórios ocultos. Para isso usamos a opção `-a`, que pode ser combinada com o `-l`. Ao usar `-la` no meu diretório verifiquei a existência do arquivo oculto “.newfile”. Os arquivos ocultos serão sempre precedidos de um “.” no nome.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls -la
total 56
drwxr-xr-x 1 jovyan jovyan 4096 Apr 28 15:21 .
drwxr-xr-x 1 jovyan jovyan 4096 Apr 28 15:10 ..
drwxr-xr-x 2 jovyan jovyan 4096 Apr 28 15:10 exomas
drwxr-xr-x 2 jovyan jovyan 4096 Apr 28 15:10 genoma
-rw-r--r-- 1 jovyan jovyan 11021 Apr 28 15:10 Homo_sapiens.GRCh38.dna.fa
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R1.fq
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R2.fq
-rw-r--r-- 1 jovyan jovyan 22 Apr 28 15:22 .newfile
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R1.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R2.fq
```

No Windows, se você quiser visualizar os arquivos ocultos é preciso alterar a guia

“Opções de pasta e pesquisa” no menu “Organizar”. Após entrar nessa guia clique em “Modo de exibição”, depois em: “Mostrar arquivos, pastas e unidades ocultas”. Nesse caso, o método do Linux é bem mais simples, pois você pode escolher na linha de comando se quer ou não enxergar os arquivos ocultos do diretório corrente.

Outras opções bastante usadas do ls são:

- -h (human format): converte os números para formatos mais fáceis de ler por humanos (MB em casos em que o tamanho do arquivo é maior que 1 MB e GB em casos que o tamanho é maior que 1 GB).
- -lt: lista os arquivos em formato longo em ordem decrescente de criação.
- -ls: lista todos os arquivos e subdiretórios em formato longo em ordem decrescente de tamanho.

Existem outras opções para o comando ls. Se você tiver interesse nesse assunto recomendo a leitura do seguinte link:

<http://www.tecmint.com/15-basic-ls-command-examples-in-linux/>.

Como faço uma cópia de um arquivo ou modifico o nome dele?

7 - cp

Com o comando cp (copy) você copia um arquivo ou diretório para outro idêntico ao original. Como exemplo, copiei o meu arquivo “pai.R1.fq” para o arquivo “pai.p1.fq” no meu próprio diretório. Após usar o comando cp, usei o comando ls -l para ver o tamanho do novo arquivo. Assim, é possível observar que o arquivo “pai.p1.fq” tem o mesmo tamanho do arquivo pai.R1.fq (3252 bytes).

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2zd0:~/curso$ cp pai.R1.fq pai.p1.fq
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls -l
total 40
drwxr-xr-x 2 jovyan jovyan 4096 Apr 28 15:10 exomas
drwxr-xr-x 2 jovyan jovyan 4096 Apr 28 15:10 genoma
-rw-r--r-- 1 jovyan jovyan 11021 Apr 28 15:10 Homo_sapiens.GRCh38.dna.fa
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R1.fq
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R2.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:30 pai.p1.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R1.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R2.fq
```

E se eu quisesse copiar o meu arquivo para outro diretório que não o atual? Eu usaria o mesmo comando e sintaxe. Neste caso, você poderia manter o nome original do seu arquivo, pois não haveria conflitos (a cópia estaria em outro diretório). No exemplo abaixo estou copiando o arquivo “pai.p1.fq” para o diretório “exomas”. Em seguida eu executo o comando `ls -l` e passando como parâmetro a pasta “exomas” e ele lista todos os arquivos desta pasta e podemos observar o arquivo copiado lá.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso$ cp pai.p1.fq exomas/
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso$ ls -l exomas/
total 12
-rw-r--r-- 1 jovyan jovyan 3249 Apr 28 15:10 filho.R1.fq
-rw-r--r-- 1 jovyan jovyan 3249 Apr 28 15:10 filho.R2.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:35 pai.p1.fq
```

8 - mv

O comando `mv` (move) é usado para mover um arquivo para outra localização no sistema. Essa operação é idêntica à operação “Recortar+Colar” do Windows. Vamos mover o arquivo “Homo_sapiens.GRCh38.dna.fa” do diretório corrente para a pasta “genoma”. Com o auxílio do `ls`, é possível verificar que o arquivo “Homo_sapiens.GRCh38.dna.fa” deixou de existir no diretório `curso`.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso$
mv Homo_sapiens.GRCh38.dna.fa genoma/
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso$
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso$ ls
exomas  genoma  mae.R1.fq  mae.R2.fq  pai.p1.fq  pai.R1.fq  pai.R2.fq
```

E se eu quisesse somente mudar o nome do meu arquivo?

No Linux você também pode usar o comando `mv` para isso. Imagine que eu quero renomear o arquivo “sequencias.fa” para “sequencias2.fa”. Para quem está acostumado com Windows pode parecer estranho, mas no fundo quando você usa a função renomear o que o Windows faz é exatamente isso: move o arquivo original para outro com o novo nome que você escolheu.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso$ mv pai.p1.fq pai.fa
```

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls
exomas  genoma  mae.R1.fq  mae.R2.fq  pai.fa  pai.R1.fq  pai.R2.fq
```

Mas, e se eu quiser criar um novo arquivo?

9 – touch

O comando touch (change file timestamps) atualiza a data e hora de acesso de um arquivo para a data e hora atuais. Porém, quando o arquivo não existe, o comando é usado para criá-lo.

Um exemplo pode ser visto logo abaixo. Nele, é possível verificar que após o uso do comando “touch mae.R1.fq” a data de modificação do arquivo (6ª e 7ª colunas) passou de Apr 28 15:10 para Apr 28 15:49. Parece inútil? Existem muitos sistemas empresariais que apagam arquivos antigos com base na data do último acesso. Dessa forma, o uso deste comando se torna importante para evitar que sejam descartados arquivos com informações importantes.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls -l
total 32
drwxr-xr-x 1 jovyan jovyan 4096 Apr 28 15:35 exomas
drwxr-xr-x 1 jovyan jovyan 4096 Apr 28 15:42 genoma
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R1.fq
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R2.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:30 pai.fa
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R1.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R2.fq
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ touch mae.R1.fq
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls -l
total 32
drwxr-xr-x 1 jovyan jovyan 4096 Apr 28 15:35 exomas
drwxr-xr-x 1 jovyan jovyan 4096 Apr 28 15:42 genoma
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:49 mae.R1.fq
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R2.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:30 pai.fa
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R1.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R2.fq
```

Para criar um novo arquivo a sintaxe é idêntica. Com a diferença que você tem que nomear seu arquivo de modo diferente dos que já existem no diretório corrente. Como

exemplo, vou criar o arquivo “painelgene.txt”. O arquivo criado está vazio, como pode ser visualizado na 5ª coluna do comando “ls -l” (0 bytes).

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ touch painelgene.txt
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls -l
total 32
drwxr-xr-x 1 jovyan jovyan 4096 Apr 28 15:35 exomas
drwxr-xr-x 1 jovyan jovyan 4096 Apr 28 15:42 genoma
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:49 mae.R1.fq
-rw-r--r-- 1 jovyan jovyan 3256 Apr 28 15:10 mae.R2.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:30 pai.fa
-rw-r--r-- 1 jovyan jovyan 0 Apr 28 15:54 painelgene.txt
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R1.fq
-rw-r--r-- 1 jovyan jovyan 3252 Apr 28 15:10 pai.R2.fq
```

Mas não quero usar o arquivo que eu acabei de criar. Existe um comando para apagá-lo?

10 – rm

O comando rm (remove) serve para apagar arquivos. Para quem está acostumado com o Windows, um **aviso importante: não existe lixeira, então é bom ter cuidado antes de sair apagando os seus arquivos e pastas**. A sintaxe é bastante simples, como mostrado abaixo:

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls
exomas  genoma  mae.R1.fq  mae.R2.fq  pai.fa  painelgene.txt  pai.R1.fq  pai.R2.fq
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ rm painelgene.txt
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ ls
exomas  genoma  mae.R1.fq  mae.R2.fq  pai.fa  pai.R1.fq  pai.R2.fq
```

Ok. Já sei criar, apagar, copiar e mover um arquivo. E se eu quiser saber o conteúdo de um arquivo, como faço?

11 – less

O comando less serve para visualizar arquivos em formato texto no Linux. Para quem usa Windows ele é parecido com o Bloco de Notas, com a diferença que não serve para alterar o conteúdo do arquivo (para isso é necessário um editor de texto). A sintaxe

dele é bastante simples, basta digitar “less nome_do_arquivo”. Assim, usei o comando less para abrir o arquivo “Homo_sapiens.GRCh38.dna.fa” que está dentro do diretório genoma. Então primeiro use o comando cd para entrar no diretório genoma e em seguida use o comando less para visualizar o conteúdo. Para sair da tela que mostra o conteúdo do arquivo basta apertar a tecla q de quit.

Dica: FASTA é um formato de armazenamento de sequências genômicas internacionalmente aceito. Cada sequência é escrita em duas ou mais linhas, onde a primeira contém o identificador (“>”), o nome da sequência e opcionalmente um comentário. As demais linhas representam os nucleotídeos ou aminoácidos da sequência. Para mais informações, recomendo o link: http://en.wikipedia.org/wiki/FASTA_format.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso$ cd genoma/
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso/genoma$
less Homo_sapiens.GRCh38.dna.fa
```

```
>chr8
GCAATTATGACACAAAAAATTAACAGTGCAGACTGATATATAAATCAAA
ACAAATGTCCTTTACATGTTTTCTGTTACAGTAGTAACAATATGTGTAAA
CTTAATTATCATATTTTTTTCTTGTGCTGTGGTTGTGCTCTGGGTTCAAT
CTCTAAAATGCTGTTACCTTAGACCAGGAGAAATATTAACCATACAGAC
TCTGTTTCAAGTCATAGCTGAATATTTCAAAAGAGTGACTTTGTAAAAA
CATGTTCCAATGGCAAATTGATTCATTGTGATGGGATCAATTATTCCAAA
```

É possível criar uma nova pasta para organizar melhor os meus dados?

12 – mkdir

O comando mkdir (make directory) permite a criação de um novo diretório. Diferentemente do Windows, onde todos os diretórios são criados com um nome padrão (“Nova pasta”), no terminal do Linux você tem que especificar no comando qual o nome do diretório que você deseja criar. Usei esse comando para criar um diretório chamado de “exoma_pai”.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso/genoma$ cd ..
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso$ mkdir exoma_pai
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2d1jc2bzd0:~/curso$ ls
exoma_pai  exomas  genoma  mae.R1.fq  mae.R2.fq  pai.fa  pai.R1.fq  pai.R2.fq
```

Reparem no primeiro comando “cd ..”. Como no comando anterior eu tinha mudado de diretório e entrado no diretório “genoma”, aqui eu voltei para o diretório anterior, ou seja, o diretório “curso”.

Tarefa: Após criar o diretório “exoma_pai” entre nele usando um comando já apresentado. Tente lembrar qual é e utilizá-lo sem auxílio do meu texto. Caso não consiga, estude melhor o comando do item 5.

Posso criar um atalho para um arquivo, como faço no Windows?

13 – ln

O comando ln é usado para criar links para arquivos. No Windows, um link para um determinado arquivo é chamado de atalho. Esse comando é útil para facilitar o acesso a um arquivo bastante utilizado e, principalmente, para evitar múltiplas cópias de arquivos grandes no sistema.

Dentro do diretório “exoma_pai”, criado no item 12, criei um link para o arquivo “pai.R1.fq”. Esse arquivo está dentro do diretório curso, que está acima do diretório atual. A sintaxe do comando é: “ln -s arquivo_origem arquivo_destino”.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ cd exoma_pai/
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/exoma_pai$
ln -s ../pai.R1.fq .
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/exoma_pai$ ls -l
lrwxrwxrwx 1 jovyan jovyan 28 Apr 28 16:16 pai.R1.fq -> /home/jovyan/curso/pai.R1.fq
```

Após a criação, perceba que o link aparece no diretório corrente em uma cor diferente e com uma indicação do caminho onde está o arquivo original. Se o link criado estivesse “quebrado” (quando o arquivo de origem não existe) a cor estaria em vermelho.

Tarefa: Visualize o arquivo pai.R1.fq usando um comando já apresentado. Tente lembrar qual é e utilizá-lo sem auxílio do meu texto. Caso não consiga, estude melhor o comando do item 11.

Perceba que eu usei alguns atalhos do sistema para executar o meu comando. Quando usei a expressão “../pai.R1.fq”, estou dizendo ao sistema que o meu arquivo de origem está no diretório pai do atual, isto é um diretório acima. Quando usei a expressão

“.”, estou dizendo ao sistema que o meu arquivo de destino deve ser gravado no diretório atual. Ficou complicado? Ok, vamos apagar o link que foi criado e executar o mesmo comando sem o primeiro atalho do sistema para facilitar o entendimento.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/exoma_pai$  
rm pai.R1.fq  
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/exoma_pai$  
ln -s /home/jovyan/curso/pai.R1.fq .  
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/exoma_pai$ ls -l  
total 0  
lrwxrwxrwx 1 jovyan jovyan 28 Apr 28 16:20 pai.R1.fq -> /home/jovyan/curso/pai.R1.fq
```

Tarefa: Visualize o arquivo “pai.R1.fq” usando o mesmo comando que você usou na tarefa anterior. Veja que não existem diferenças em relação ao que você tinha visto quando criou o link usando o atalho do sistema.

Na verdade, os comandos são iguais. Se você utilizar o comando `pwd`, perceberá que o diretório “/home/jovyan/curso” é o diretório pai do diretório corrente que é “/home/jovyan/curso/exoma_pai”. Dessa forma, o caminho completo dele pode ser substituído pelo atalho do sistema “.”.

Dica: No Linux a sintaxe para manipulação de arquivos, na maioria das vezes, se dá no formato origem/destino. Por exemplo, os comandos `cp`, `mv`, `ln`.

14 – wc

O `wc` (word count) é outro comando bastante interessante para manipulação de arquivos no Linux. Com ele é possível saber quantas linhas e caracteres possui um arquivo texto qualquer. A sintaxe do comando e o resultado do mesmo para o arquivo `sequencia2.fa` (o link que foi criado no item anterior).

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/exoma_pai$  
wc pai.R1.fq  
51 52 3252 pai.R1.fq
```

Podemos ver que, para o arquivo `pai.R1.fq`, o comando `wc` retornou três números (51, 52 e 3252). O primeiro diz respeito ao número de linhas no arquivo, o segundo ao número de palavras e o terceiro ao número de bytes (ou caracteres).

15 – grep

O comando `grep` serve para buscar, em um ou mais arquivos, por linhas que contenham um determinado padrão. Por exemplo, no item 11 (comando `less`) utilizamos um arquivo que contém sequências em formato FASTA. Considerando que nesse formato, a primeira linha de cada sequência contém o sinal “>” seguido do nome da mesma, seria possível obter os nomes das sequências deste arquivo sem sequer abri-lo? É para isso que é utilizado o `grep`. Basta buscar pelo padrão “>” e teremos todas as linhas que nomeiam as sequências.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/exoma_pai$ cd ..
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso$ cd genoma/
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/genoma$
grep ">" Homo_sapiens.GRCh38.dna.fa
>chr8
```

E se eu quiser saber quantas sequências tem esse arquivo, posso usar o `grep` também?

Para fazer isso, usamos o mesmo comando, com a diferença que agora queremos contar o número de linhas onde o padrão que procuramos ocorre. Para isso a sintaxe é praticamente a mesma. Basta inserir a opção “-c” antes do nome do arquivo. Podemos observar que esse arquivo só tem uma sequência.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/genoma$
grep -c ">" Homo_sapiens.GRCh38.dna.fa
1
```

E se o programa que eu executei for demorar dias pra rodar? Terei que esperar até o fim para sair da máquina?

16 – Nohup e &

Quando um programa vai demorar bastante tempo para terminar (como por exemplo um programa que faça alinhamento de sequências), é interessante, ao mesmo tempo, usarmos o “&” para liberar o terminal e o “nohup”, que coloca o processo para rodar *background*, isto é, desvincula a execução do programa do seu desktop. Desse modo,

podemos usar o terminal para outras tarefas (função do “&”) e garantimos que o programa vai ser executado no servidor remoto ainda que você feche seu terminal (função do “nohup”).

Para simular esse comportamento vamos usar o comando abaixo chamado sleep. Esse comando deixa o terminal em modo de espera pela quantidade de segundo que você especificar. No exemplo abaixo, sleep 20, deixa o terminal esperando por 20 segundos.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/genoma$ sleep 20
```

O seu terminal vai ficar “preso” aguardando o sleep terminar por 20 segundos. Agora se você usar o comando com nohup e & você vai ver que seu terminal é liberado para que você possa executar outras tarefas e assim que ele terminar você verá uma mensagem que aquele processo foi concluído.

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/genoma$ nohup sleep 20  
&
```

```
[1] 124
```

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/genoma$ nohup:  
ignoring input and appending output to 'nohup.out'
```

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/genoma$
```

```
[1]+ Done nohup sleep 20
```

```
jovyan@jupyter-osvaldoreisss-2dlinux-5fcourse-2dljc2bzd0:~/curso/genoma$
```