



## Linux

Vagner Okura
LaCTAD
vagnerko@unicamp.br



## Por que Linux para Bioinformática?



- Linux é GRATUITO (Open Source/Free), e a maioria das suas ferramentas também;
- A maioria das aplicações de Bioinformática são desenvolvidas para Linux;
- Fácil controle de processos/uso de recursos;
- Amplamente utilizado pela comunidade científica;
- Fácil modificação para interesses específicos;
- Excelente suporte para scripting e programação;
- Excelente suporte para clusterização, multiprocessamento, computação distribuída.



## Qual distribuição Linux utilizar?



Arch	CentOS	Debian
OpenSUSE	Gentoo	Linux Mint
Fedora	Red Hat	Slackware
SUSE	Ubuntu	BioLinux

- Servidores de grande porte:
   recomendação para
   distribuições comerciais (Red
   Hat, SUSE) devido a
   estabilidade, confiabilidade
   (\$\$ para suporte e garantia),
   OU, distribuições derivadas
   (CentOS, OpenSUSE).
- Desktop: a critério do usuário.
- BioLinux: vem com vários pacotes de bioinformática instalados. Roda em PCs, e como máquina virtual.



### Como trabalhar com sistema Linux?

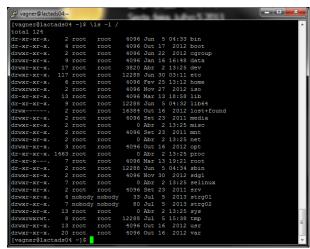


- Shell é a interface dos usuários com o sistema operacional Linux.
- Shell é um interpretador de linguagem de comandos, o qual executa comandos recebidos de um dispositivo entrada padrão (teclado) ou arquivo.
- Usuários interagem com um Shell Linux usando um terminal.













### Qual shell utilizar?



- Há diferentes interpretadores de comando em Linux:
  - sh, csh, ksh, tcsh, bash
- Utilizaremos o bash
  - Histórico de comandos
    - Navegação no histórico: setas para cima e para baixo



- Busca comandos por palavra:
  - CTRL+R mkdir -> busca o último comando com a palavra "mkdir";
  - CTRL+R mkdir CTRL+R -> busca o penúltimo comando com a palavra "mkdir".
- Edição de linha de comando
- Empilhamento de diretórios (pushd , popd)
- "Completamento" de linha de comando (tecla TAB)
  - Busca parcialmente nomes de programas, arquivos e variáveis



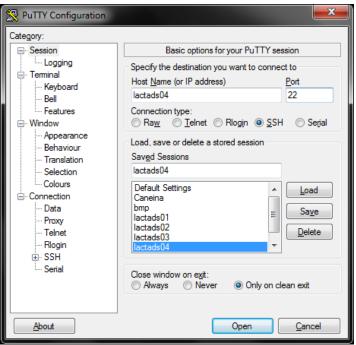
#### Como acessar remotamente um servidor Linux?

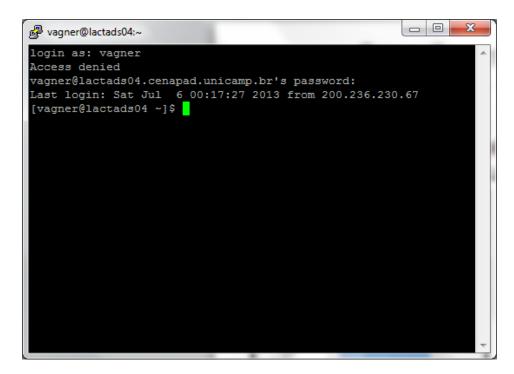
#### Windows -> Linux



### putty

- Implementação (free) de ssh para Windows.
- Faz conexão com servidor Linux, emulando terminal Linux.







#### Como acessar remotamente um servidor Linux?

Linux -> Linux



#### ssh - secure shell:

- Ferramenta para login remoto e seguro: autenticação e criptografia
- Permite execução de comandos remotos
- Ex:
  - ssh -l joana 193.6.37.12: conecta (loga) no servidor com IP 193.6.37.12 com o usuário "joana".
  - ssh -p 2222 jorge@server1.unicamp.br: conecta (loga) no servidor "server1.unicamp.br" com o usuário "jorge".



## Ajuda de comandos/programas



- man <comando>: fornece informações detalhadas do comando especificado;
- <comando> -h: muitos comandos (e softwares de bionformática) possuem uma documentação própria, que pode ser acessada por meio de um parâmetro. Pode haver diversas variações, como: --h, -help, --help;



## Ajuda de comandos/programas



• Exemplo: man cp

```
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
       --backup[=CONTROL]
              make a backup of each existing destination file
       -b
             like --backup but does not accept an argument
```



### Wildcards



- Os caracteres '\*' e '?' tem uso especial.
- '\*': corresponde a nenhum ou mais caracteres em um nome de arquivo ou diretório.
  - ls list\*
    - => Listará todos os arquivos no diretório atual, cujo nome começa com "list".
  - ls \*list
    - => Listará todos os arquivos no diretório atual, cujo nome termina com "list".
- '?': corresponde a um caracter.
  - ls ?ouse
    - => Listará arquivos como 'house 'and 'mouse', mas não 'grouse'.
  - ls ?????.txt
    - => Listará arquivos cujo nome tenha cinco caracteres e extensão ".txt".
  - ls bac\*.???
    - => Listará arquivos cujo nome comece com a palavra "bac" e termine com três caracteres após o ponto ".



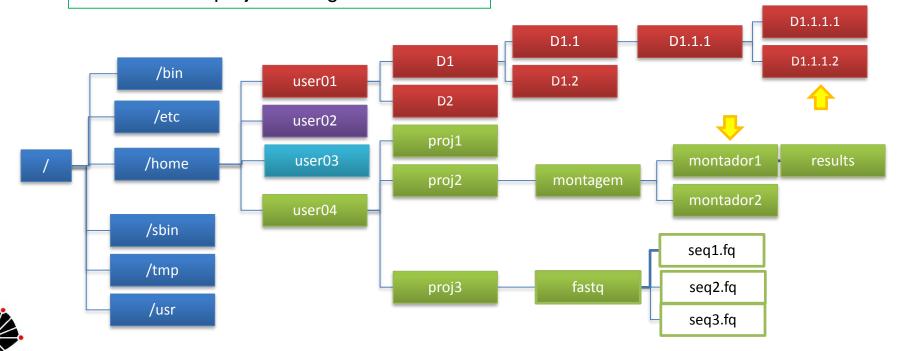


- pwd: mostra diretório atual
  - pwd significa "print working directory".
  - Exemplos:

UNICAMP

/home/user01/D1/D1.1/D1.1.1/D1.1.1.2

/home/user04/proj2/montagem/montador1





#### • cd: muda de diretório

Comando	Descrição
cd /proj1	muda para o diretório "proj1" a partir da raiz
cd proj1	muda para o diretório "proj1" a partir do diretório atual
cd proj2/montagem/montador1	muda para o diretório especificado a partir do diretório atual
cd /	muda para o diretório raiz
cd	muda para o diretório home
cd ~	muda para o diretório home
cd	muda para o diretório um nível acima do atual
cd/	muda para o diretório dois níveis acima do atual
cd//proj3/montagem	muda para o diretório proj3/montagem, dois níveis acima do atual





### • Is: lista conteúdo de um diretório

Comando	Descrição
ls	lista conteúdo do diretório atual
ls .	lista conteúdo do diretório atual
ls -l	lista conteúdo do diretório atual usando formato de listagem
12 -1	longa
ls proj1	lista conteúdo do diretório "proj1" a partir do diretório atual
ls proj2/montagem/montador1	lista conteúdo do diretório especificado a partir do diretório
15 projz/moncagem/moncadori	atual
ls /	lista conteúdo do diretório raiz
ls ~	lista conteúdo do diretório home
ls -1	lista conteúdo do diretório um nível acima do atual usando
	formato de listagem longa





### mkdir/rmdir: cria e remove diretórios

Comando	Descrição
mkdir proj1	cria diretório "proj1" no diretório atual
mkdir proj2 proj3	cria diretórios "proj1" e "proj2" no diretório atual
mkdir proj1/montagem	cria diretório "montagem" dentro do diretório "proj1" (diretório "proj1" deve ter sido criado anteriormente).
mkdir -p proj1/montagem	cria diretório "montagem" dentro do diretório "proj1" (diretório "proj1" não precisa ter sido criado anteriormente).
rmdir proj1	remove diretório "proj1" do diretório atual. Diretório "proj1" deve estar vazio para ser removido com o comando mkdir.
rmdir proj2 proj3	remove diretórios "proj1" e "proj2" do diretório atual. Diretórios devem estar vazios para serem removidos com o comando mkdir.





### • cp: copia arquivos e diretórios

Comando	Descrição
cp arq1 arq2	cria uma cópia do arquivo "arq1" com o nome "arq2"
cp chr1.fasta chr2.fasta proj1	copia arquivos "chr1.fasta" e "chr2.fasta" (do diretório atual) para o diretório "proj1" (diretório "proj1" deve ter sido criado anteriormente).
cp proj1/* proj2	copia todos os arquivos do diretório "proj1" para o diretório "proj2" (diretório "proj2" deve ter sido criado anteriormente).
cp -r proj1 proj2	copia todos os arquivos e/ou diretórios (recursivamente) do diretório "proj1"para o diretório "proj2".  Se diretório "proj2" não existe, faz cópia de "proj1" no diretório atual.  Se diretório "proj2" foi criado anteriormente, "proj2/proj1".
cp *fasta/proj3	copia arquivos e/ou diretórios, a partir do diretório atual, cujo nome termine com a palavra "fasta", para o diretório "proj3" que está um nivel acima.





### • rm: remove arquivos e diretórios

Comando	Descrição
rm arq1 arq2	remove arquivos "arq1" e "arq2" do diretório corrente.
rm proj1/chr1.fasta	remove arquivo "chr1.fasta" do diretório "proj1".
rm proj2/*	remove todos os arquivos do diretório "proj2".
rm -r proj2	remove diretório "proj2", com todos os arquivos e/ou diretórios (recursivamente).
rm -r *	remove todos os arquivos e diretórios do diretório atual recursivamente). Certifique-se em que diretório está antes de executar este comando!!!





### • mv: move/renomeia arquivos e diretórios.

Comando	Descrição
mv arq1 arq2	renomeia arquivo "arq1" para "arq2" no diretório corrente.
mv chr1.fasta chr2.fasta proj1	move arquivos "chr1.fasta" e "chr2.fasta" do diretório atual para o diretório "proj1" (diretório "proj1" deve ter sido criado anteriormente).
mv *fasta proj2	move arquivos e/ou diretórios, do partir do diretório atual, cujo nome termine com a palavra "fasta", para o diretório "proj2" (diretório "proj1" deve ter sido criado anteriormente).
mv dir1 dir2	move diretório "dir1" do partir do diretório atual para o diretório "dir2" (diretórios "dir1" e "dir2" devem ter sido criados anteriormente ["dir2/dir1"] ). Se "dir2" não existir, diretório "dir1" é renomeado para "dir2".
mv ~/*.txt dir3	move arquivos com extensão ".txt" do diretório home para o diretório "dir3" (diretório "dir3" deve ter sido criado anteriormente).



### Comandos para controle de processos



### • ps/kill: lista informações/mata processos

Comando	Descrição
ps	lista processos sendo executados na sessão da shell
ps -f	lista processos sendo executados na sessão da shell, no formato completo
ps -e	lista todos os processos sendo executados no servidor
ps -ef	lista todos os processos sendo executados no servidor, no formato completo
ps -fu user01	lista todos os processos do usuário "user01" sendo executados no servidor, no formato completo
kill -kill 1234	mata o processo com PID "1234".
kill -9 1234 6767	mata os processos com PID "1234" e "6767"



### Comandos para controle de processos



top: exibe tabela em tempo real dos processos.

Tasks: 5921 total, 1 running, 5920 sleeping, 0 stopped, 0 zombie Cpu(s): 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st 2051122284k total, 1144370760k used, 906751524k free, 338012k buffers Swap: 7812024k total, 788k used, 7811236k free, 1095416848k cached PID USER VIRT RES SHR S %CPU %MEM TIME+ COMMAND NI 131997 vagner.o 0 19540 5844 1000 R 0.0 0:03.48 top 20 8.5 0:01.87 ksoftirgd/193 584 root 20 0 0 5 0.3 0.0 1138 root 0 5 0.3 0.0 0:34.59 events/175 20 0 1156 root 20 0 0 5 0.3 0.0 0:26.43 events/193 6509 root 0 5 0.3 0.0 3:10.04 jbd2/md0-8 20 14094 root 20 0 195m 7476 996 S 0.3 0.0 103:52.87 snmpd 14100 root 20 248m 3968 2116 5 0.3 0.0 39:43.25 lsi mrdsnmpagen 15385 root 0 46.3g 1.2g 10m 5 0.3 0.1 43:47.06 java 20 15839 root 0 44.9g 111m 10m 5 0.3 0.0 9:50.96 java 20 0 19356 1576 1264 5 0.0 0.0 0:30.16 init 20 1 root 2 root 20 0 0 5 0.0 0.0 0:01.56 kthreadd 0 0:25.67 migration/0 3 root RT0 5 0.0 0.0 0:51.13 ksoftirgd/0 4 root 20 0 0 5 0.0 0.0 0:00.00 migration/0 5 root RT0 0 5 0.0 0.0 0 5 0.0 0.0 0:10.79 migration/1 6 root 0 RT0:00.00 migration/1 RT0 5 0.0 0.0 7 root 0 5 0.0 0.0 0:05.49 ksoftirgd/1 8 root 20 0 5 0.0 0.0 0:08.00 migration/2 RT9 root



## Comandos para controle de processos



### • &/nohup: executa processos em background

Comando	Descrição
run assembiv.beri &	executa programa "run_assembly.perl", liberando o prompt do shell.
nohup run_assembly.perl &	executa programa "run_assembly.perl", liberando o prompt do shell. Caso haja perda de conexão do terminal com o servidor, o processo continuará sendo executado.



### Comandos para visualizar arquivos texto



### • cat/more/less: mostra conteúdo de arquivos texto

Comando	Descrição
cat report.txt	mostra todo o conteúdo do arquivo "report.txt" na tela. Se tiver muitas linhas, excederá o buffer do terminal. Usado principalmente para concatenar arquivos.
more report.txt	mostra conteúdo do arquivo "report.txt", paginando conforme o tamanho da tela. Barra de espaço movimenta para a próxima pagina.
less report.txt	similar ao comando "more", mostra conteúdo do arquivo "report.txt", mas permite movimentar o arquivo para frente e para trás.  - teclas Seta Cima / Seta Baixo: movimenta arquivo uma linha para cima/baixo.  - teclas PageUp / PageDown: movimenta arquivos uma pagina para cima/baixo.  - ESC+'>': vai para o final do arquivo  - ESC+'<': vai para o início do arquivo  - '/' <palavra>: busca palavra no arquivo a partir do ponto que está até o final do arquivo. Palavra é destacada na tela.  - '?' <palavra>: busca palavra no arquivo a partir do ponto que está até o início do arquivo. Palavra é destacada na tela.</palavra></palavra>

### Comandos para editar arquivos texto



- nano: edita arquivo texto
  - Nano

GNU nano 2.0.9

Arquivo: sequencias.fa

gi | 627728334 | gb | KJ472479.1 |

ATCTTTCACTTGGGAACGTGTTTGGTGTAAGATGGGAAAGCCACGTGGATTGCGTACTGCTCGTAAGCAT GTGAATCATCGCCGTGAACAGAGATGGGCTGATAATGATTACAAGAAAGCACATCTTGGCACACGTTGGA AGGCGAATCCTTTTGGTGGTGCATCCCATGCCAAAGGAATTGTTTTAGAAAAAGTAGGAGTAGAAGCTAA TTTGTTCCACGAGATGGTTGTTTGAATTATTGAAGAGAATGATGAAGTCCTGGTGGCAGGCTTTGGTC GTAAAGGTCATGCTGTGGGAGATATTCCTGGTGTTCGTTTCAAGGTAGTGAAAGTTGCAAACGTTTCATT ACTTGCATTGTATAAAGAGAAGAAAGAAGACCAAGATCATAATCTGTGTATTGTCATGTTGAATAAACA 

>gi|12963252|gb|AF321248.1|

GGGCGAGGGGCCCGCCTCTTCGATTCCACCGGGAAGATGCCCAAACTTGATCATTTAGAGGAAGTAAAA GTCGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTAACGTGTGTCTTGAATCACGCTTAAAA AAAAAAAAAAAAAGAGAATAAATCTCACAATGAGAATACACAAAAGACGGGCCGAACCCGCAAGTGGTG GTGGGTCACACCCCGTCGCGAGCGGTACGCTTCCCCTCCACCGAGCGGGGAGTGCGTCGGAGTCCCGCT GGTTGAACGAGAGGAAAGCGACGCCGGGTACTGTGCATTAACGGTTTAAAGTCACCTGAAATCGCCCGGC CACTCGCAAAGGGTTACTCAAATCCCCTCCAGTTCCGCGCGGGAAGCAAGAACTGCATTCCTGCCCCGCG TGCGGGGGCGTGGATGTGCCCTGTACACAATAAAAACTATTGAGAACGTAACGCTGAGACCTGTATGGTA CTCTGAAAACTAAAAGTACATAACCCTGAACGGTGGATCACTCGGCTCGTGGGTCGATGAAGAACGCAGC AAATTGCGCGCCGACATGTGAACTGCAGGACACATGAACATCGACATTTCGAACGCACATTGCGGTCCTT

 $^G = Ctrl+G$ 



^G Ajuda











## Comandos para compressão de arquivos



### zip, unzip; gzip, gunzip; bzip2, bunzip2

possuem diferentes taxas de compressão

possitem and entes taxas ac compressas	
Comando	Descrição
zip archive.zip arq1.fa arq2.fa	cria arquivo compactado "archive.zip" com os arquivos "arq1.fa" e "arq2.fa"
zip archive.zip arq*.fa	cria arquivo compactado "archive.zip" com os arquivos cujo nome inicia com "arq" e termina dom ".fa".
zip -r archive.zip proj1	cria arquivo compactado "archive.zip" com todos o arquivos e diretórios (recursivamente) do diretório "proj1"
unzip archive.zip	descompacta o conteúdo do "archive.zip" no diretório atual.
gzip arq1.fa arq2.fa	cria arquivos compactados "arq1.fa.gz" e "arq2.fa.gz"
gzip arq*.fa	cria arquivos compactados com extensão ".gz" para os arquivos cujo nome inicia com "arq" e termina dom ".fa".
gzip -r proj1	cria arquivos compactados com extensão ".gz" para todos os arquivos (recursivamente) do diretório "proj1".
gunzip arq1.fa.gz arq2.fa.gz	descompacta os arquivos "arq1.fa.gz" e "arq2.fa.gz"
gunzip proj2/*.gz	descompacta os arquivos com extensão ".gz" do diretório "proj2"
bzip2 arq1.fa arq2.fa	cria arquivos compactados "arq1.fa.bz2" e "arq2.fa.bz2"
bunzip2 arq1.fa.bz2 arq2.fa.bz2	descompacta arquivos "arq1.fa.bz2" e "arq2.fa.bz2"



## Comandos para manipulação de arquivos



## • grep: busca por padrões em arquivos

Comando	Descrição
grep "gene" arq.txt	busca no arquivo "arq.txt" linhas com a palavra "gene"
grep "^gene" arq.txt	busca no arquivo "arq.txt" linhas que iniciam com a palavra "gene"
grep "gene\$" arq.txt	busca no arquivo "arq.txt" linhas que terminam com a palavra "gene"
grep -c "gene" arq.txt	busca no arquivo "arq.txt" linhas com a palavra "gene" e imprime número de linhas encontradas



### Comandos para manipulação de arquivos



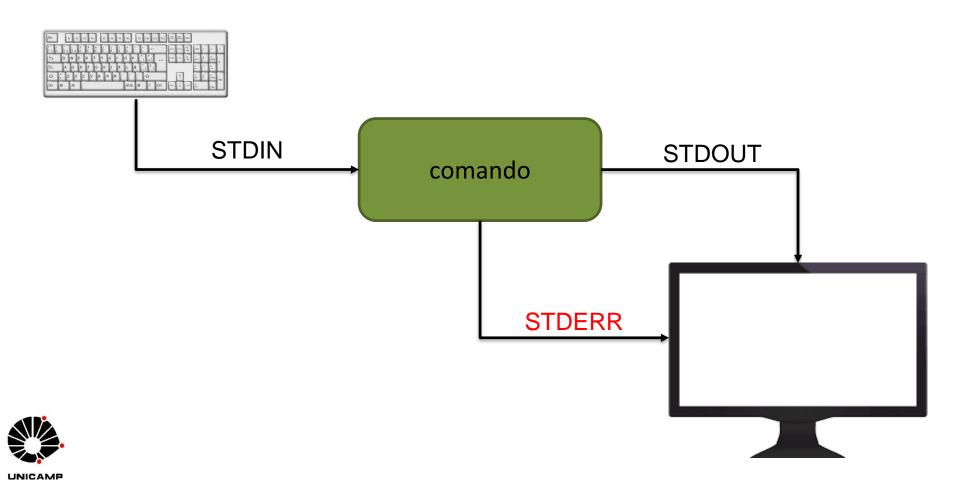
- sort: ordena linhas de um arquivo
- uniq: filtra linhas repetidas de um arquivo
- head: imprime as primeiras linhas ou os primeiros caracteres de um arquivo
- tail: imprime as últimas linhas ou os últimos caracteres de um arquivo
- wc: conta número de linhas, palavras e bytes de um arquivo



### Redirecionamento e comandos aninhados



Redirecionamento de dados (stdout/stderr/stdin)



#### Redirecionamento e comandos aninhados



### Redirecionamento de dados (stdout/stderr/stdin)

Comando	Descrição
history > commands_history.txt	direciona saída padrão (stdout) do comando "history" para o arquivo "commands_history.txt" (se arquivo já existir, sobrescreve conteúdo)
cat arq1.fa arq2.fa > arq_all.fa	cria arquivo "arq_all.fa" com conteúdo dos arquivos "arq1.fa" e "arq2.fa"
cat arq3.fa >> arq_all.fa	concatena conteúdo do arquivo "arq3.fa" ao arquivo "arq_all.fa"
rm -r proj1 2> log.txt	direciona saída de erro (stderr) do comando "rm" para o arquivo "log.txt" (se arquivo já existir, sobrescreve conteúdo)
rm -r proj2 2>> log.txt	concatena a saída de erro (stderr) do comando "rm" para o arquivo "log.txt"
sort < lista.txt	ordena o conteúdo do arquivo "lista.txt", recebido como entrada padrão (stdin). O comando "sort" recebe o conteúdo do arquivo "lista.txt".



#### Redirecionamento e comandos aninhados



- Comandos aninhados: PIPE (|):
  - Permite enviar saída de um comando diretamente como entrada de outro comando.
    - # cat arq1.txt arq2.txt | grep gene
      - => o comando "cat" concatena o conteúdo dos arquivos "arq1.txt" e "arq2.txt", direcionando a saída, via o Pipe, para o comando "grep", que busca linhas com a palavra "gene". Exibe todas as linhas dos arquivos "arq1.txt" e "arq2.txt" em que aparece a palavra "gene".
    - # sort arq1.txt | uniq | wc -l
      - => o comando "sort" ordena o arquivo "arq1.txt", direcionando a saída, via o Pipe, para o comando "uniq", que filtra as linhas repetidas, direcionando a saída, via o Pipe, para o comando "wc", que conta o número de linhas. Conta o número de entradas únicas do arquivos "arq1.txt"



### Comando para fazer download de arquivos



- wget
  - Suporte para HTTP, HTTPS, FTP;
  - wget http://www.lactad.unicamp.br/downlods/arquivol.fa.gz

