

Introduction to Linux

EU-US Bioinformatics Short Course

June 2013

Newark, Delaware?

Username

rna1	rna11	rna21	rna31
rna2	rna12	rna22	rna32
rna3	rna13	rna23	rna33
rna4	rna14	rna24	rna34
rna5	rna15	rna25	rna35
rna6	rna16	rna26	rna36
rna7	rna17	rna27	rna37
rna8	rna18	rna28	rna38
rna9	rna19	rna29	rna39
rna10	rna20	rna30	rna40

Login (ssh) to Biohen

```
$ ssh -X user@biohen.dbi.udel.edu
```

Password



RespectYourElders

Changing your password

```
$ passwd
```

```
Changing password for user dnasko.
```

```
Enter login(LDAP) password:
```

```
New password:
```

```
New password:
```

Who am I ?

```
$ whoami
```

- Prints your user name (who you are)

Where am I ?

\$ pwd

- Prints your **p**resent **w**orking **d**irectory

List Directory Contents

\$ ls

- Prints the contents of your current directory (may be blank)

List Directory Contents

```
$ ls /home/microb3
```

- Prints the contents of your current directory (may be blank)

Command History

Try pressing [arrow up]

- Then edit your previous command

Command Arguments

```
$ ls -l /home/microb3
```

```
$ ls -lh /home/microb3
```

- These are called arguments, or parameters, they change the behavior of the command.

Manual

```
$ man ls
```

- This will display the help for the 'list' command.
- Scroll with [up / down], press q to exit.

Changing the directory

```
$ cd /home/microb3
```

- Now list the contents

Changing the directory

```
$ cd ~
```

- Changes you back to your home directory
- Same as cd /home/username

Creating a symbolic link

```
$ ln -s /home/microb3 ~/microb3-shared
```

```
$ ls -l
```

Make a new directory

```
$ mkdir test
```

- Now change into that directory and type:

```
$ pwd
```

Make a new directory

```
$ cd ../
```

- To get back to your original directory

Copy a file

```
$ cp [from] [to]
```

```
$ cp ~/microb3-shared/file1.txt ./
```

```
$ cp ~/microb3-shared/file2.txt ./
```

- `./` is how Linux refers to the current directory

Move a file

```
$ mv [from] [to]
```

```
$ mv fi ← Then hit [tab]
```

```
Now hit [tab][tab]
```

Wild card

*

```
$ mv file* test/
```

- Moves anything that starts with “file” to the directory called test.

Rename a file

- Change into your `test/` directory

```
$ mv file1.txt file-one.txt
```

- Indeed, the **`mv`** command is used to both move and rename files.
- Yeah, it's weird. But it makes sense.

File permissions

```
$ ls -l
```

```
[dnasko@biohen test]$ ls -l
```

```
total 8
```

```
-rw-r--r-- 1 dnasko wommack 27 Jun 17 10:35 file2.txt
-rw-r--r-- 1 dnasko wommack 27 Jun 17 10:35 file-one.txt
```

Permissions

Owner

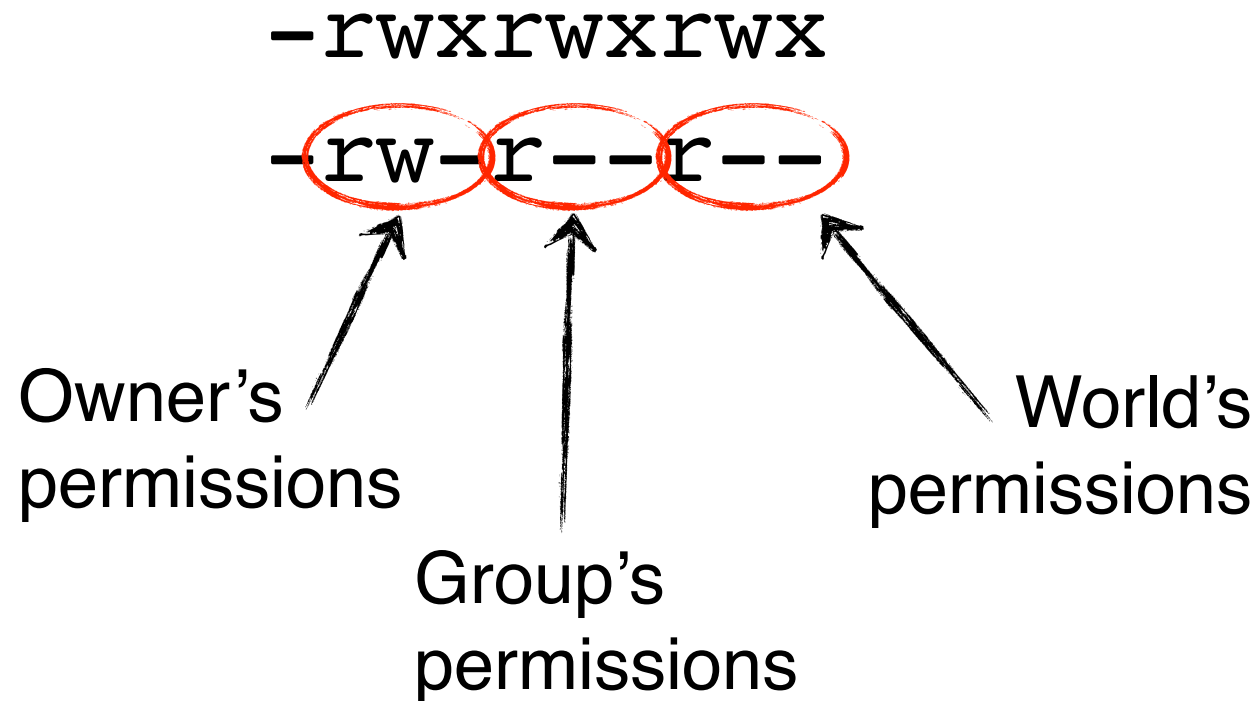
Owner's
Group

Date
Modified

File size
(in Bytes)

File name

File permissions



File permissions

r	w	x	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

644

-rw-r--r--

File permissions

\$ groups

- See what `groups` you are in.

File permissions

```
$ chmod 600 file-one.txt
```

```
$ ls -l
```

- Change permissions so only you can read and write

File permissions

```
$ chmod 644 file-one.txt
```

```
$ ls -l
```

- Change permissions back so only you can read and write, but your group and the world can still read.

Redirecting

```
$ ls -l
```

Redirecting

```
$ ls -l > list.out
```

```
$ cat list.out
```

- The greater than sign redirects whatever was going to go to the screen to a file.

Redirecting

```
$ ls -l >> list.out
```

```
$ cat list.out
```

- Two greater than signs appends whatever was going to go to the screen to a file.

Redirecting

```
$ ls -l >> list.out
```

```
$ cat list.out
```

- Two greater than signs appends whatever was going to go to the screen to a file.

Piping

```
$ ls -l | grep "file"
```

- Pipe allows you to feed the output of one command as the input of a new command.
- `grep` allows for searching

Piping

```
$ ls -l | wc
```

- Pipe allows you to feed the output of one command as the input of a new command.
- `wc` prints the newline, word and byte counts.

Deleting file

```
$ rm -i file2.txt
```

iNO Undo!

Deleting directory

```
$ cd ../
```

```
$ rmdir test
```

- rmdir only removes empty directories!!

Deleting directory

```
$ rm -i test/file-one.txt
```

```
$ rm -i test/list.out
```

```
$ rmdir test
```

- There we go.

Try this . . .

- Create a folder called `microb3-work` under your home directory.
- Copy all the files in: `~/microb3-shared/raw_sequences/` to the new directory you just created

Path

- Deviation from the workbook, I'll correct this. -- Not using `tophat` for this example.

Redirecting

```
$ metagene
```

- You'll receive an error because **metagene isn't in your PATH**

Path

```
$ echo $PATH
```

- Displays the directories where programs are located.
- Check out: bioinformatics.udel.edu/Core/BioHen-Software

Path

```
$ less /home/microb3/class_env
```

- Allows you to open and view a file.
- No editing allowed.
- Great because this will not read file into memory.

Edit a file

```
$ nano ~/.bashrc
```

- Add this line to the bottom of the file:

```
source /home/microb3/class_env
```

[ctrl-o] to save; [enter] confirm file name

[ctrl-x] to exit

Path

- Any command in your `.bashrc` is run automatically when you log in.

```
$ exit
```

- And now log back in.

Path

\$ metagene

- Works!

Unzip a file (gzip format)

```
$ cd ~/microb3-work
```

```
$ gunzip *.gz
```

- Change to the directory you just copied sequence files to in the previous exercise.

View the file

```
$ less FL1-1.fastq
```

- A brief discussion on FASTQ

FASTQ

Line 1	@	Sequence Header
Line 2		Sequence
Line 3	+	Comment Line
Line 4		Quality Scores

```
@HWI-ST741:112:C03J8ABXX:6:2103:11138:184506 1:N:0:GCCAAT
GGTGGCAGGAATTCATTACTGCTGTCTCTTTCTTGGTTGTCATCTGCCTGCAGCTTGCAGTCTTGGCAACACTTCTTTAGCTTGTTTACTGCTTCTTTAG
+
@C@DFFFFHHHHJJIGIHIIJJIDEHHIIGHGGCGHFHIJGDGGGBB@FHIDIJIGG<FGIACCCG@D=>A>?=) ).7;B6.;A(65(-;;5;AC###
@HWI-ST741:112:C03J8ABXX:6:2106:20786:184584 1:N:0:GCCAAT
CAATTTGCTTACTTGCTGTTGCACCTGCAGCAGGAGCTTGACCCGCTCGATGGGCGCTACCGCCGTCTTGCTTATGGGGGGGGGGGGGGCGCCCCCCCCG
+
@@@A?BDFAHFGHIIIGIFHIGHIIIGEHE9EC?CGGAFHGGGBHHGFGG4@C#####
```

Quality Scores

```
@HWI-ST741:112:C03J8ABXX:6:2103:11138:184506 1:N:0:GCCAAT
GGTGGCAGGAATTCATTACTGCTGTCTCTTCTTGGTTGTCATCTGCCTGCAGCTTGCAGTCTTGGCAACACTTCTTTAGCTTGTTTACTGCTTCTTTTAG
+
@C@DFFFFHHHHJJJIGIHIIJJIDEHHIIGHGGCGHFHIJGDGGGBB@FHIDIJIGG<FGIACCCG@D=>A>?=) ).7;B6.;A(65(-;;5;AC###
```

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Quality Scores

$$Q = 64$$

$$64 - 33 = 31$$

- What does that mean ?

$$1/10^{3.1} = 0.0007943282$$

- So a 0.079% chance that that base call is wrong.

Torque PBS

List Torque nodes

```
$ pbsnodes
```

List current jobs

```
$ qstat -aln
```

Start an interactive Torque session

```
$ qsub -N intrctv-users1 -I -V -l  
nodes=1:ppn=1:class
```

```
$ qstat -aln
```

Start an interactive Torque session

- To end the interactive session:

```
$ exit
```

Alias

```
$ less /home/microb3/class_env
```

- As a result, you need only type:

```
$ int
```

Batch mode

```
$ less ~/microb3-shared/bin/test.qs
```

- To submit this batch script:

```
$ qsub ~/microb3-shared/bin/test.qs
```

- Monitor your job with:

```
$ qstat -aln
```

Batch mode

- Open the result files with less:

```
$ less Test.o#####
```


Batch mode

- Delete an active queued job

```
$ qsub ~/microb3-shared/bin/test.qs
```

```
$ qstat -aln
```

```
$ qdel #####
```