

Electronic Computer Center
Al-Nahrain University
Linux Module
A. Altaher
Lecture 7 & 8



- Command line (Terminal)



Displaying a file

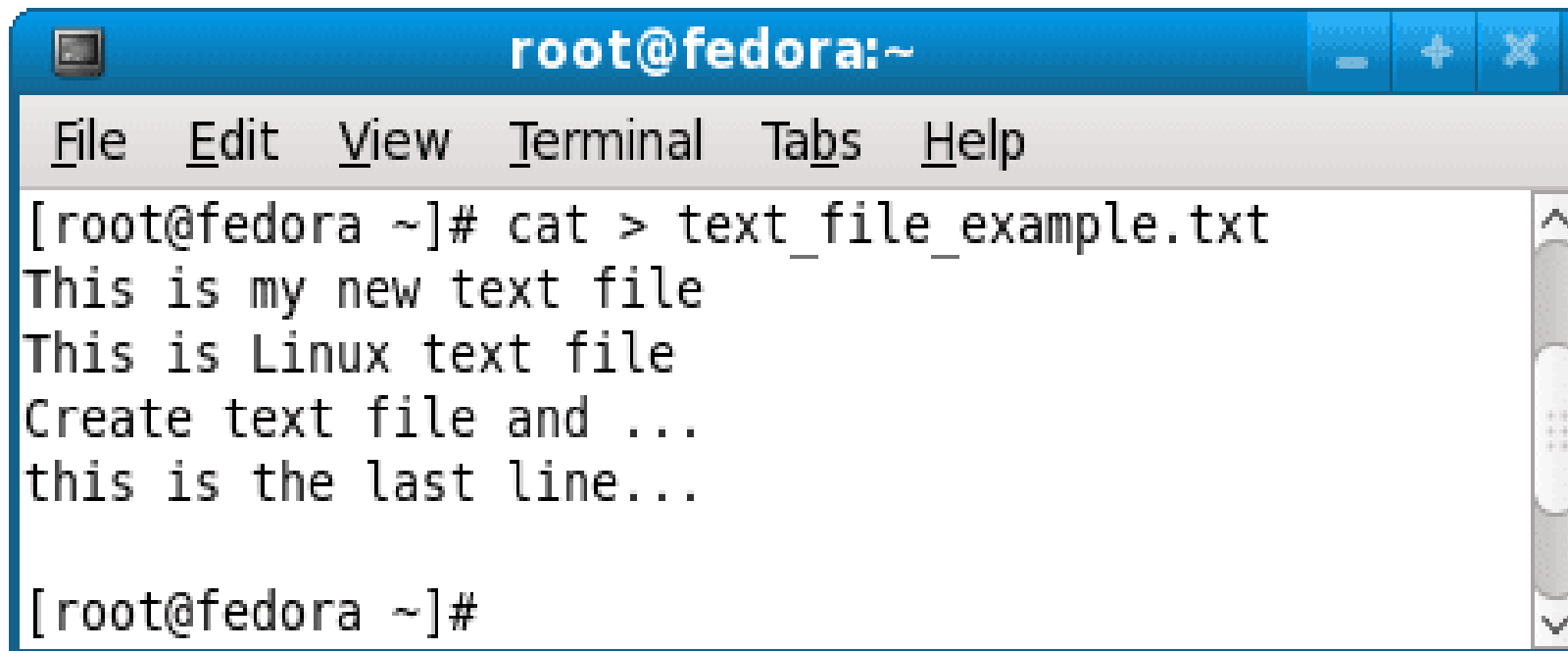
Various ways to display a file in Unix

- cat
- less
- head
- tail



Command: cat

- Dumps an entire file to standard output
- Good for displaying short, simple files

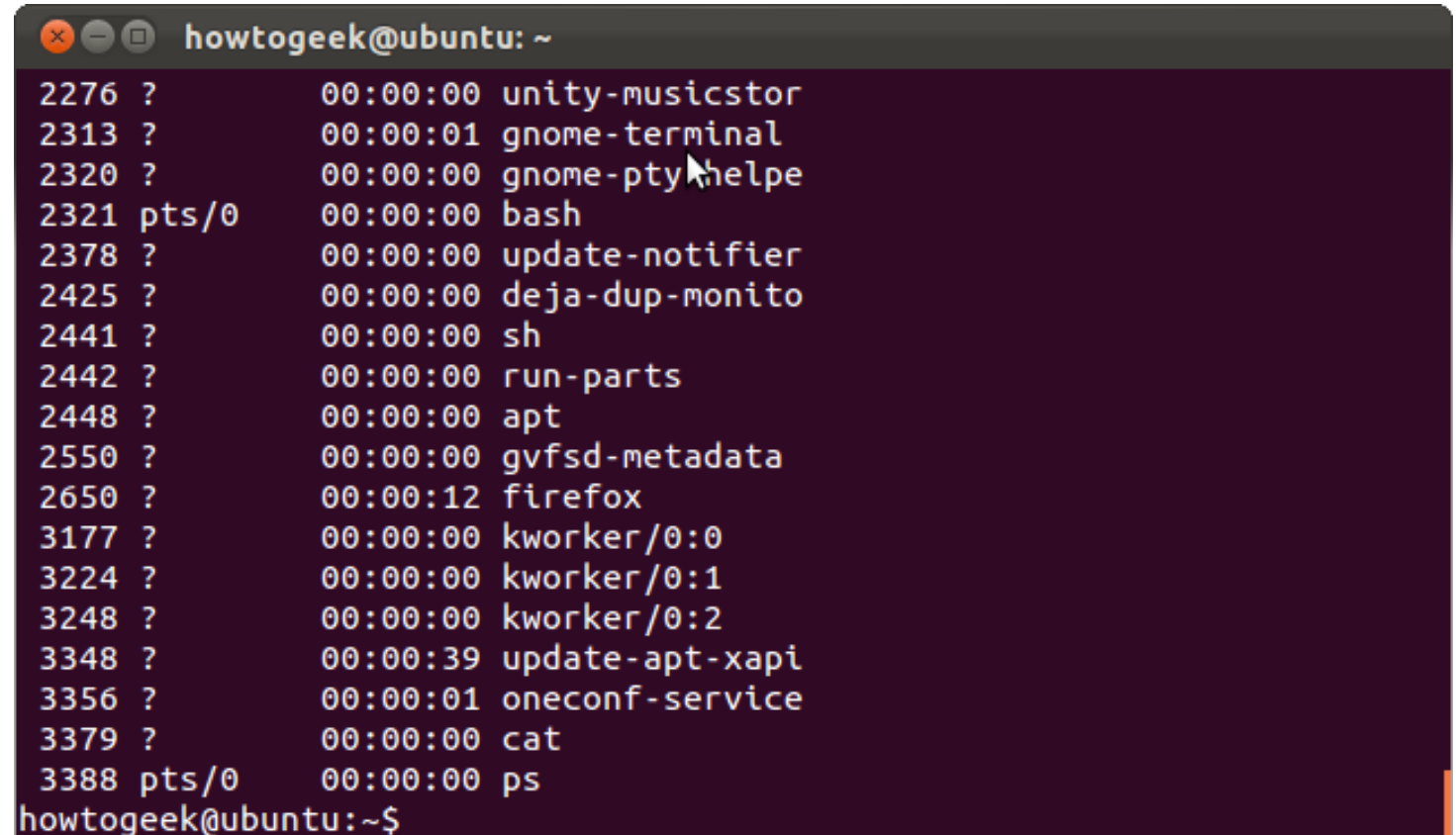


```
root@fedora:~  
File Edit View Terminal Tabs Help  
[root@fedora ~]# cat > text_file_example.txt  
This is my new text file  
This is Linux text file  
Create text file and ...  
this is the last line...  
[root@fedora ~]#
```



Command: less

- less” displays a file, allowing forward/backward movement within it
 - return scrolls forward one line, space one page
 - y scrolls back one line, b one page
- use “/” to search for a string
- Press q to quit



```
howtogeek@ubuntu: ~  
2276 ?      00:00:00 unity-musicstor  
2313 ?      00:00:01 gnome-terminal  
2320 ?      00:00:00 gnome-ptyhelpe  
2321 pts/0   00:00:00 bash  
2378 ?      00:00:00 update-notifier  
2425 ?      00:00:00 deja-dup-monito  
2441 ?      00:00:00 sh  
2442 ?      00:00:00 run-parts  
2448 ?      00:00:00 apt  
2550 ?      00:00:00 gvfsd-metadata  
2650 ?      00:00:12 firefox  
3177 ?      00:00:00 kworker/0:0  
3224 ?      00:00:00 kworker/0:1  
3248 ?      00:00:00 kworker/0:2  
3348 ?      00:00:39 update-apt-xapi  
3356 ?      00:00:01 oneconf-service  
3379 ?      00:00:00 cat  
3388 pts/0   00:00:00 ps  
howtogeek@ubuntu:~$
```



Command: head

- “head” displays the top part of a file
- By default it shows the first 10 lines
- -n option allows you to change that
- “head -n50 file.txt” displays the first 50 lines of file.txt

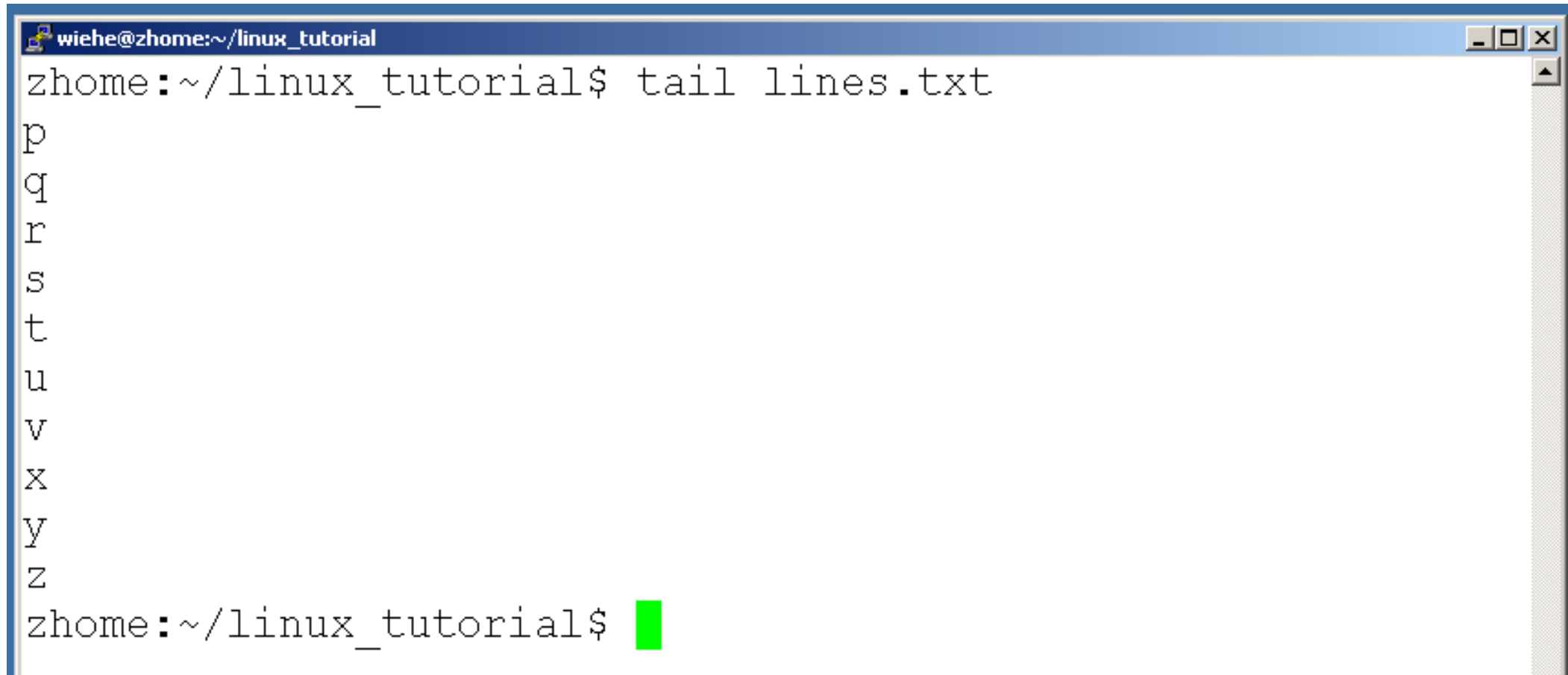


```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ head lines.txt
a
b
c
d
e
f
g
h
i
j
zhome:~/linux_tutorial$
```

Linux Terminal Commands

Command: tail

Same as head, but shows the last lines

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The terminal shows the command 'tail lines.txt' being executed. The output consists of the letters 'p', 'q', 'r', 's', 't', 'u', 'v', 'x', 'y', and 'z' on separate lines. The prompt 'zhome:~/linux_tutorial\$' is followed by a green cursor block.

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ tail lines.txt
p
q
r
s
t
u
v
x
y
z
zhome:~/linux_tutorial$
```



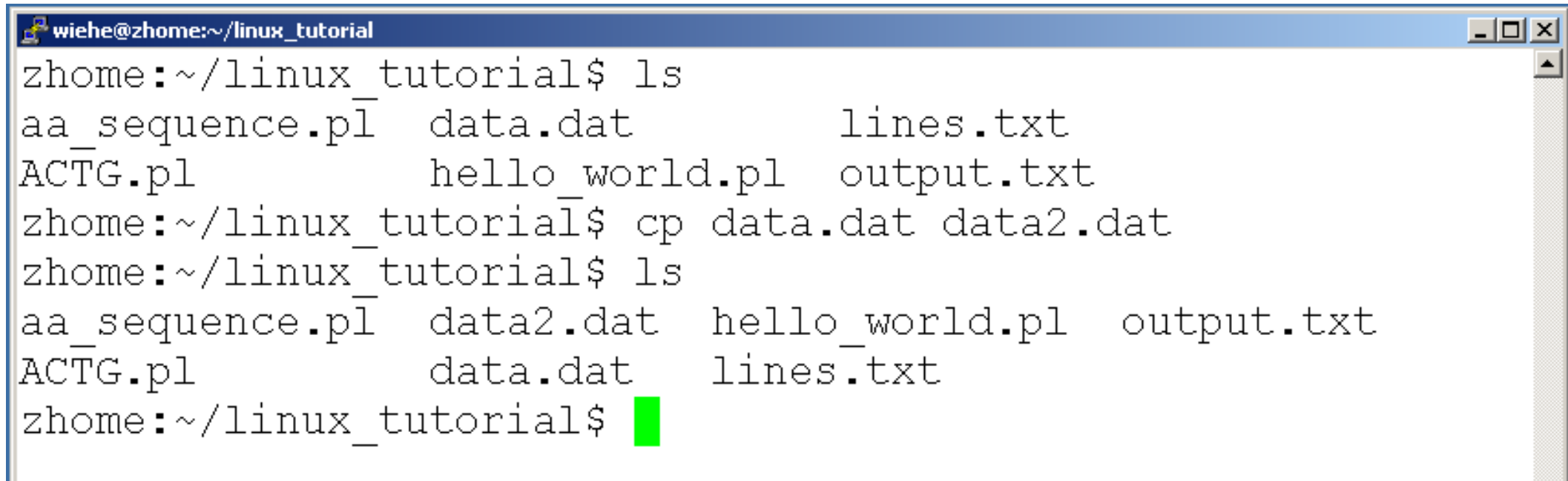
Linux Terminal Commands

- **cp (copy)** To copy a file use “cp”

\$ cp source-file(s) destination

To copy entire directories (including their contents), use a recursive copy:

\$ cp -rd source-directories destination-directory

A screenshot of a Linux terminal window. The title bar shows the user 'wiehe' at host 'zhome' in the directory '~/linux_tutorial'. The terminal content shows a sequence of commands and their outputs. First, 'ls' is run, listing files: 'aa_sequence.pl', 'data.dat', 'lines.txt', 'ACTG.pl', 'hello_world.pl', and 'output.txt'. Then, 'cp data.dat data2.dat' is executed. Finally, 'ls' is run again, showing the updated directory contents: 'aa_sequence.pl', 'data2.dat', 'hello_world.pl', 'output.txt', 'ACTG.pl', 'data.dat', and 'lines.txt'. The prompt is currently at 'zhome:~/linux_tutorial\$' with a green cursor.

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls
aa_sequence.pl  data.dat      lines.txt
ACTG.pl        hello_world.pl output.txt
zhome:~/linux_tutorial$ cp data.dat data2.dat
zhome:~/linux_tutorial$ ls
aa_sequence.pl  data2.dat    hello_world.pl  output.txt
ACTG.pl        data.dat     lines.txt
zhome:~/linux_tutorial$
```

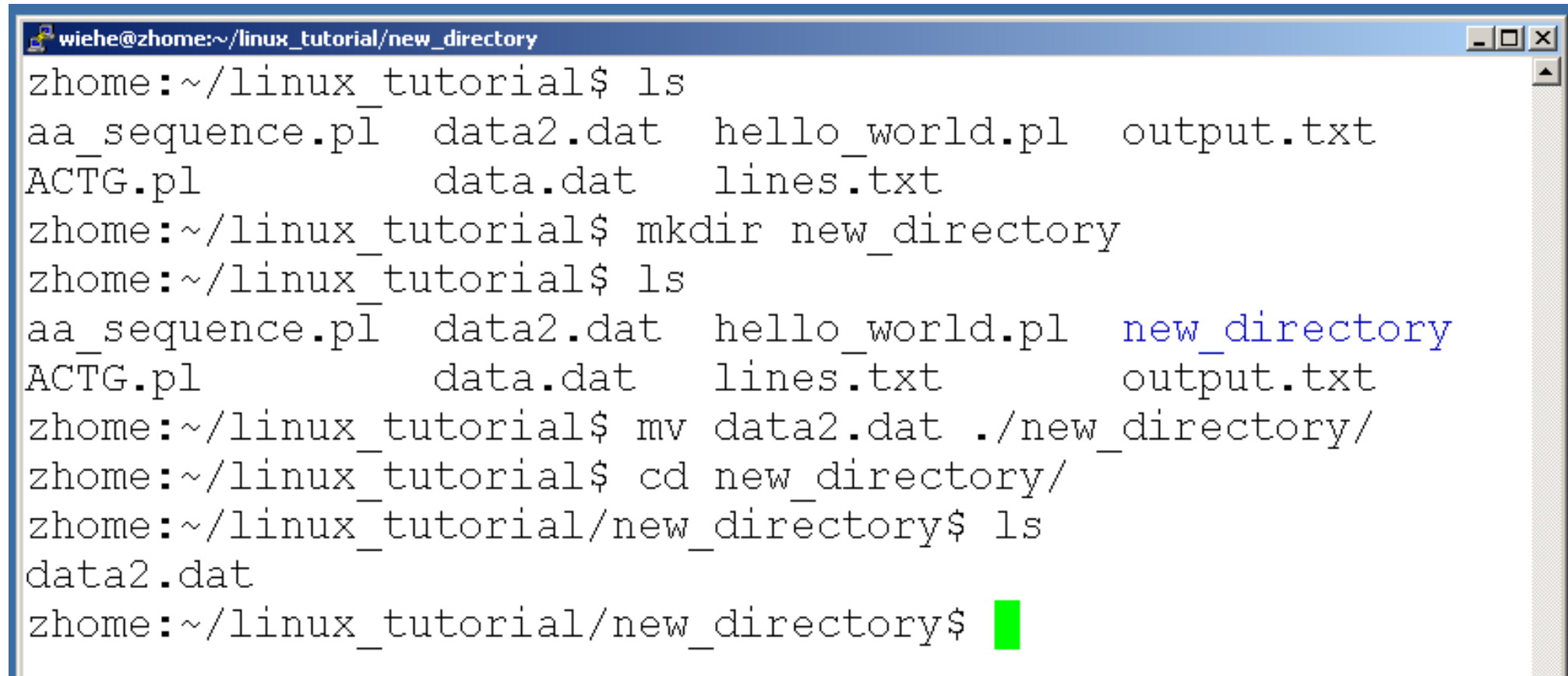


Linux Terminal Commands

- **mv (move/rename)**

\$ mv source destination

is used to rename files/directories and/or move them from one directory into another. Exactly one source and one destination must be specified.

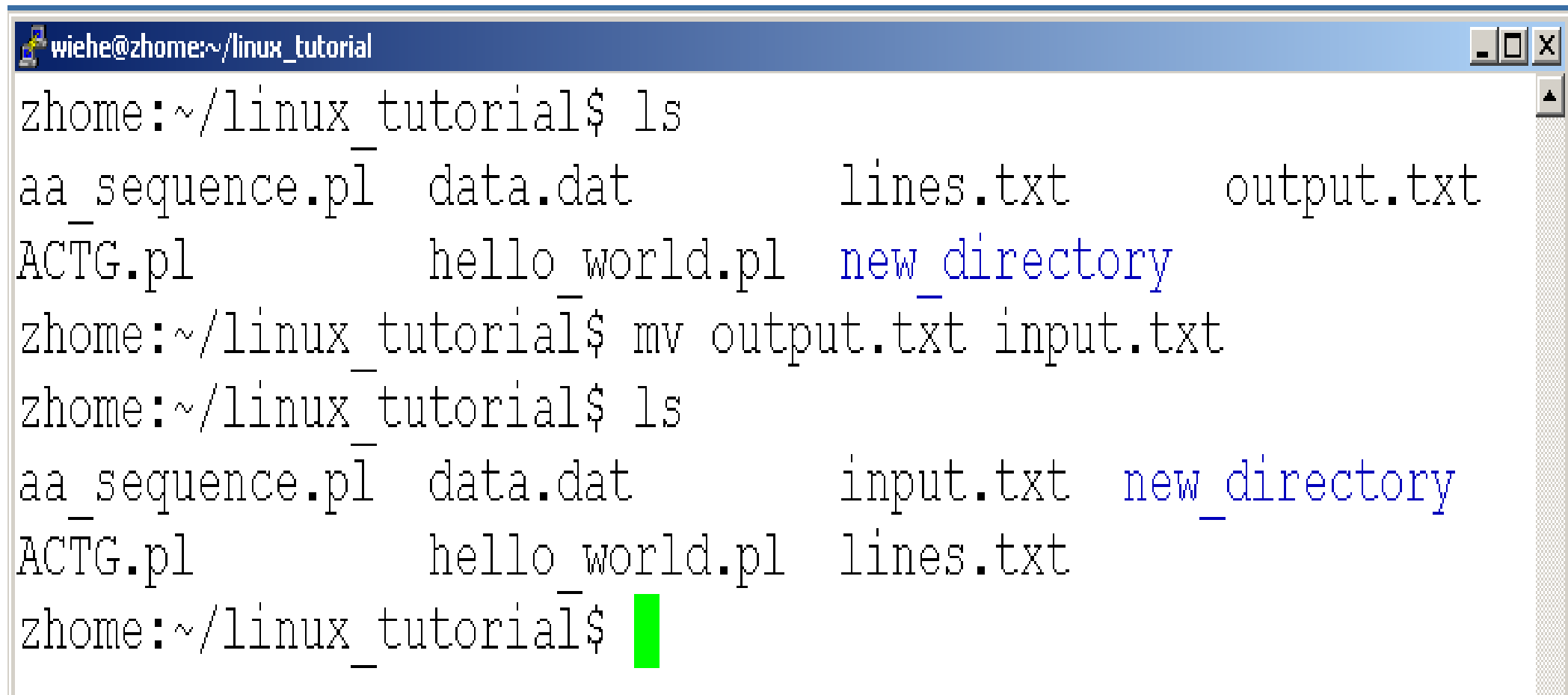
A screenshot of a Linux terminal window. The title bar shows the user 'wiehe' at host 'zhome' in the directory '~/linux_tutorial/new_directory'. The terminal shows a sequence of commands and their outputs. First, 'ls' is run in the home directory, listing several files. Then, 'mkdir new_directory' is run to create a new subdirectory. Another 'ls' command shows the new directory has been created. Then, 'mv data2.dat ./new_directory/' is run to move the file. Finally, 'cd new_directory/' is run to change into the new directory, and a final 'ls' command shows 'data2.dat' has been moved there. The prompt is currently at 'zhome:~/linux_tutorial/new_directory\$' with a green cursor.

```
wiehe@zhome:~/linux_tutorial/new_directory
zhome:~/linux_tutorial$ ls
aa_sequence.pl  data2.dat  hello_world.pl  output.txt
ACTG.pl        data.dat   lines.txt
zhome:~/linux_tutorial$ mkdir new_directory
zhome:~/linux_tutorial$ ls
aa_sequence.pl  data2.dat  hello_world.pl  new_directory
ACTG.pl        data.dat   lines.txt       output.txt
zhome:~/linux_tutorial$ mv data2.dat ./new_directory/
zhome:~/linux_tutorial$ cd new_directory/
zhome:~/linux_tutorial/new_directory$ ls
data2.dat
zhome:~/linux_tutorial/new_directory$
```



- **mv (rename)**

\$ mv source destination



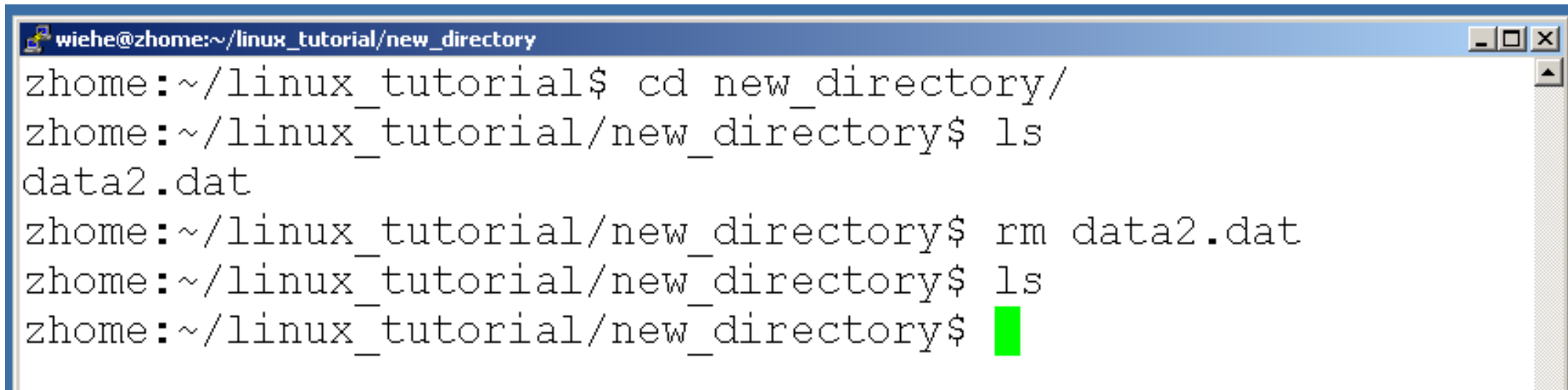
```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls
aa_sequence.pl  data.dat      lines.txt     output.txt
ACTG.pl        hello_world.pl new_directory
zhome:~/linux_tutorial$ mv output.txt input.txt
zhome:~/linux_tutorial$ ls
aa_sequence.pl  data.dat      input.txt     new_directory
ACTG.pl        hello_world.pl lines.txt
zhome:~/linux_tutorial$
```

- **rm (remove/delete)**

To remove a file use “rm”

\$ rm target-file(s) (works as shift delete)

-r (recursive) -f (force) -rf (forces deleting everything)

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial/new_directory'. The terminal shows a sequence of commands and their outputs: 'cd new_directory/' is executed, followed by 'ls' which outputs 'data2.dat'. Then 'rm data2.dat' is executed, followed by another 'ls' command. The final prompt is followed by a green cursor. The window has standard Linux window controls (minimize, maximize, close) in the top right corner.

```
wiehe@zhome:~/linux_tutorial/new_directory
zhome:~/linux_tutorial$ cd new_directory/
zhome:~/linux_tutorial/new_directory$ ls
data2.dat
zhome:~/linux_tutorial/new_directory$ rm data2.dat
zhome:~/linux_tutorial/new_directory$ ls
zhome:~/linux_tutorial/new_directory$
```



- **rm (remove/delete)**

\$ rm target-file(s) (works as shift delete)

-r (recursive) -f (force) -rf (forces deleting everything)

To remove a file “recursively”: rm -r

Used to remove all files and directories

Be very careful, deletions are permanent in Unix/Linux

- **Sudo** Superuser



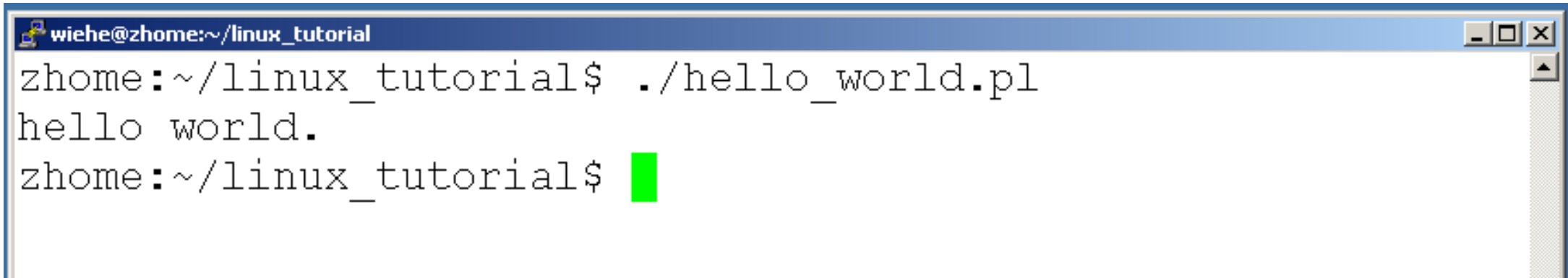
Running a program (a.k.a. a job)

Make sure the program has executable permissions

Use “./” to run the program

e.g:

Running the sample perl script “hello_world.pl”

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The terminal shows a command prompt 'zhome:~/linux_tutorial\$' followed by the command './hello_world.pl'. The output 'hello world.' is displayed on the next line. The prompt 'zhome:~/linux_tutorial\$' is shown again on the third line, followed by a green cursor block.

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ./hello_world.pl
hello world.
zhome:~/linux_tutorial$
```



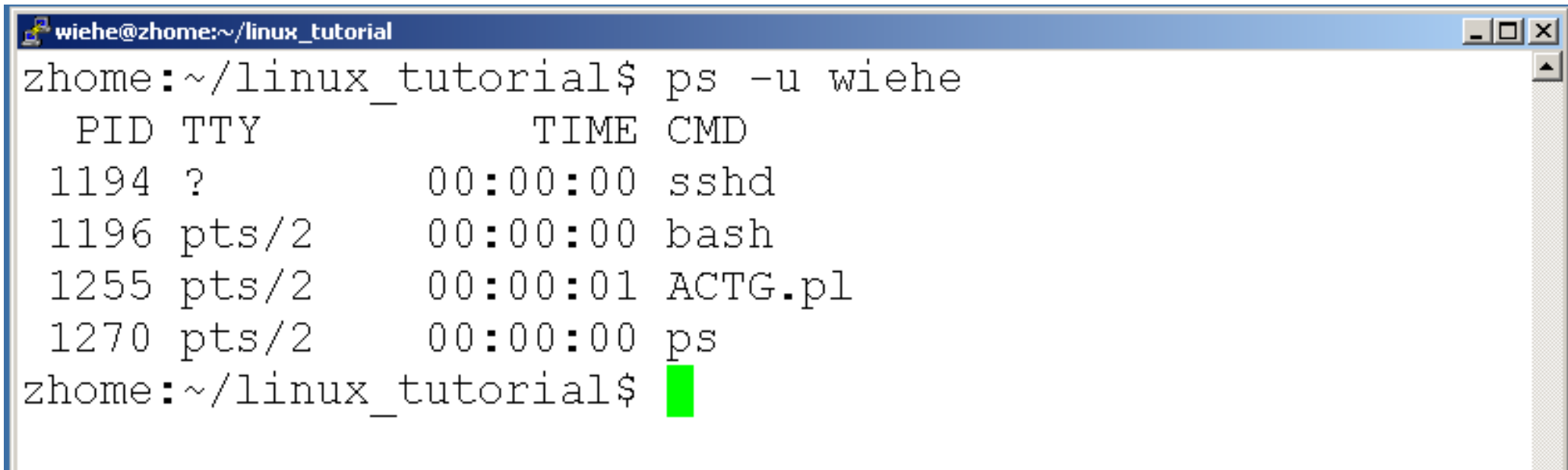
Ending a program

To end a program use “ctrl-c”. To try it:



Command: ps

To view the processes that you're running:



```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ps -u wiehe
  PID TTY          TIME CMD
 1194 ?            00:00:00 sshd
 1196 pts/2        00:00:00 bash
 1255 pts/2        00:00:01 ACTG.pl
 1270 pts/2        00:00:00 ps
zhome:~/linux_tutorial$
```

Command: top

To view the CPU usage of all processes:

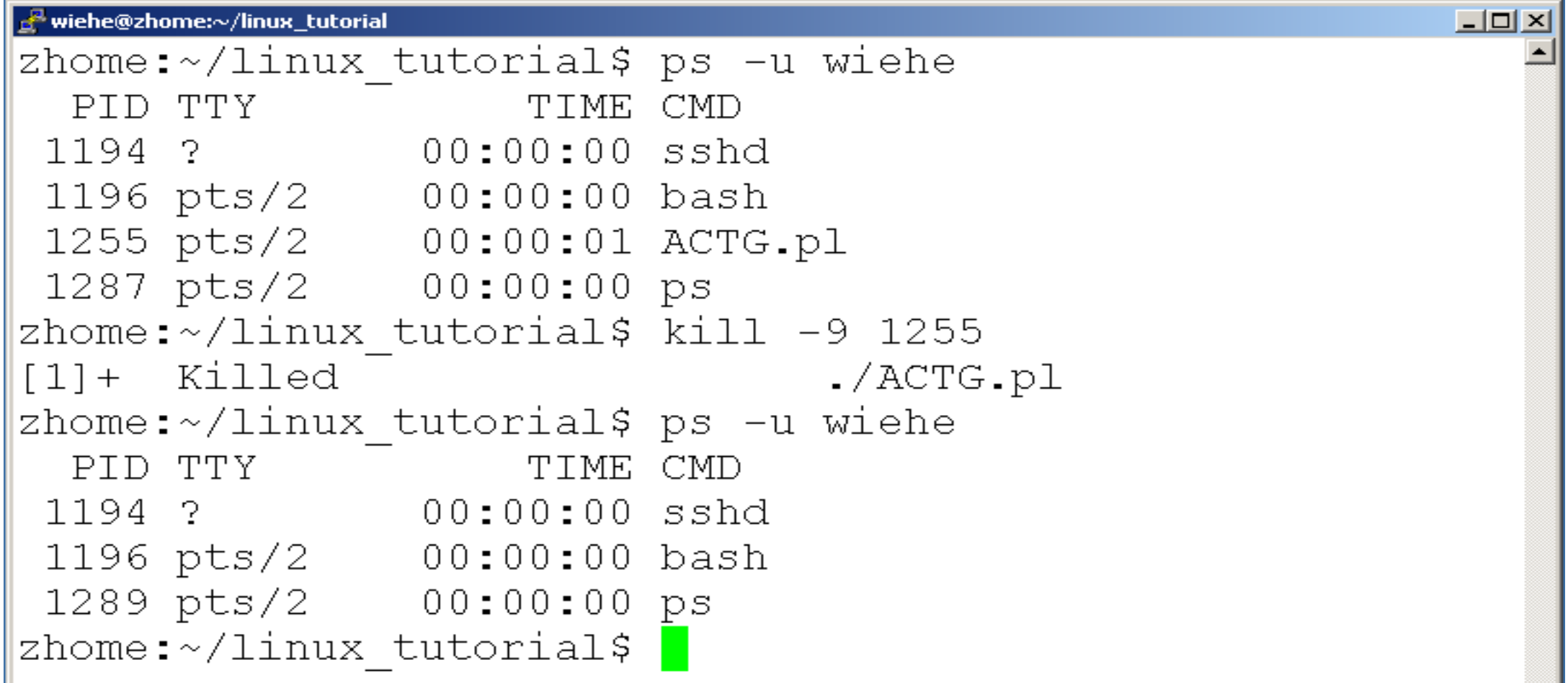
```
wiehe@zhome:~/linux_tutorial
top - 13:46:33 up 50 days,  4:26,  2 users,  load avera
Tasks:      total,      running,      sleeping,      stoppe
Cpu(s) :      us,      sy,      ni,      id,      w
Mem:      total,      used,      free,
Swap:      total,      used,      free,

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM
3403	root	15	0	0	0	0	S	0.7	0.0
1	root	16	0	1604	324	292	S	0.0	0.0
2	root	RT	0	0	0	0	S	0.0	0.0
3	root	34	19	0	0	0	S	0.0	0.0
4	root	RT	0	0	0	0	S	0.0	0.0
5	root	34	19	0	0	0	S	0.0	0.0
6	root	RT	0	0	0	0	S	0.0	0.0
7	root	34	19	0	0	0	S	0.0	0.0
8	root	RT	0	0	0	0	S	0.0	0.0
9	root	34	19	0	0	0	S	0.0	0.0

Command: kill

To terminate a process use “kill”

A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing the execution of the 'kill' command. The user first runs 'ps -u wiehe' to list processes. Then, they run 'kill -9 1255' to terminate the process with PID 1255. The terminal shows '[1]+ Killed ./ACTG.pl' as feedback. Finally, they run 'ps -u wiehe' again to confirm the process is gone.

```
wiehe@zhome:~/linux_tutorial$ ps -u wiehe
  PID TTY          TIME CMD
 1194 ?            00:00:00 sshd
 1196 pts/2        00:00:00 bash
 1255 pts/2        00:00:01 ACTG.pl
 1287 pts/2        00:00:00 ps
wiehe@zhome:~/linux_tutorial$ kill -9 1255
[1]+  Killed                  ./ACTG.pl
wiehe@zhome:~/linux_tutorial$ ps -u wiehe
  PID TTY          TIME CMD
 1194 ?            00:00:00 sshd
 1196 pts/2        00:00:00 bash
 1289 pts/2        00:00:00 ps
wiehe@zhome:~/linux_tutorial$
```

Input/Output Redirection (“piping”)

- Programs can output to other programs
- Called “piping”
- “program_a | program_b”
 - program_a’s output becomes program_b’s input
- “program_a > file.txt”
 - program_a’s output is written to a file called “file.txt”
- “program_a < input.txt”
 - program_a gets its input from a file called “input.txt”



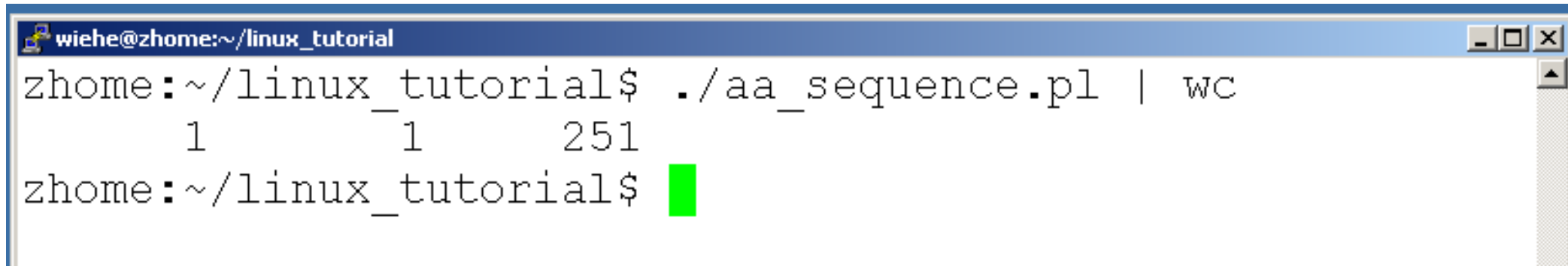
A few examples of piping

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls
aa_sequence.pl  hello_world.pl  new_directory
ACTG.pl        input.txt
data.dat       lines.txt
zhome:~/linux_tutorial$ ./aa_sequence.pl > sequence.txt
zhome:~/linux_tutorial$ ls
aa_sequence.pl  hello_world.pl  new_directory
ACTG.pl        input.txt       sequence.txt
data.dat       lines.txt
zhome:~/linux_tutorial$ less sequence.txt
```



Command: wc

- To count the characters, words, and lines in a file use “wc”
- The first column in the output is lines, the second is words, and the last is characters

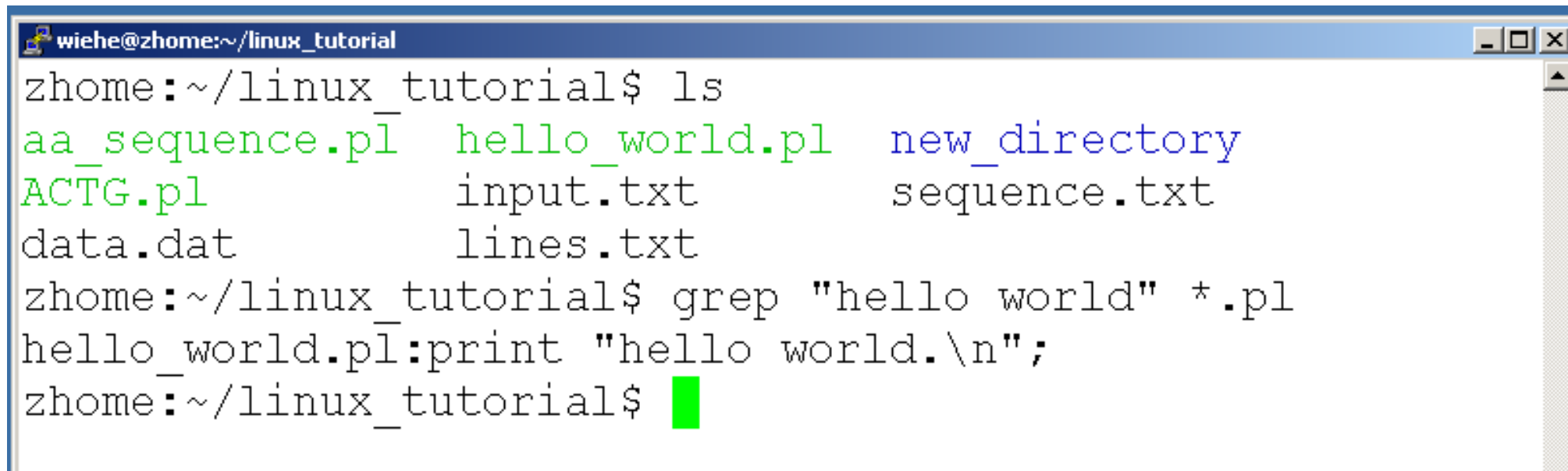


```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ./aa_sequence.pl | wc
      1      1     251
zhome:~/linux_tutorial$
```



Command: grep

To search files in a directory for a specific string use “grep”



```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls
aa_sequence.pl  hello_world.pl  new_directory
ACTG.pl         input.txt       sequence.txt
data.dat        lines.txt
zhome:~/linux_tutorial$ grep "hello world" *.pl
hello_world.pl:print "hello world.\n";
zhome:~/linux_tutorial$
```

Command: diff

To compare to files for differences use “diff”

Try: `diff /dev/null hello.txt`

`/dev/null` is a special address -- it is always empty, and anything moved there is deleted



ssh, scp

ssh is used to securely log in to remote systems, successor to telnet

`ssh [username]@[hostname]`

Try:

`ssh yourusername@localhost`

Type “exit” to log out of session

Scp is used to copy files to/from remote systems, syntax is similar to cp:

`scp [local path] [username]@[hostname]:[remote file path]`

Try:

`scp hello.txt yourusername@localhost:scp-test.txt`



Youtube

https://www.youtube.com/watch?v=9t_gJWC32zk

Linux Handout & Tutorial

<http://www.guru99.com/unix-linux-tutorial.html>

William Knottenbelt Imperial college London 2001

<http://www.doc.ic.ac.uk/~wjk/UnixIntro/index.html>

WORLD OF ASIC 2014

<http://www.asic-world.com/scripting/unix3.html>



<http://www.ee.surrey.ac.uk/Teaching/Unix/>

<http://www.ugu.com/sui/ugu/show?help.beginners>

<http://en.wikipedia.org/wiki/Unix>

