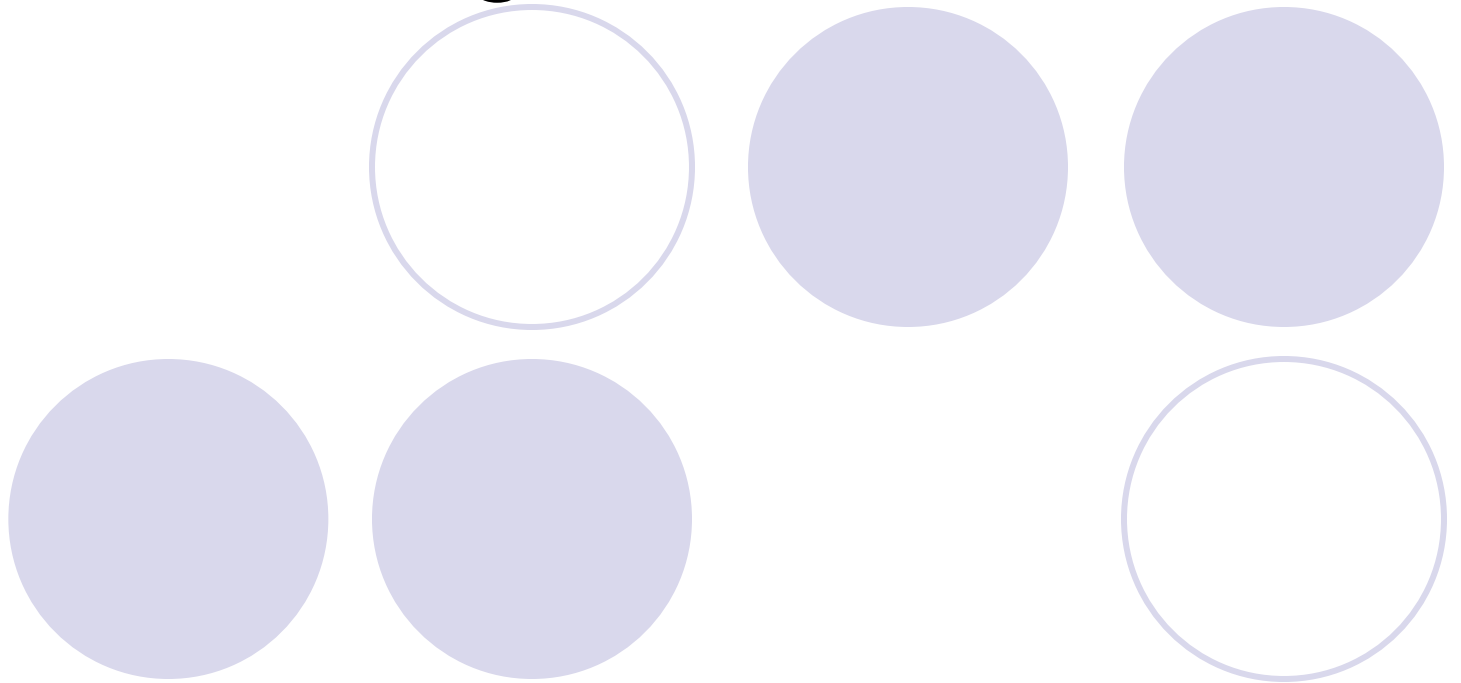


Learning Unix/Linux



Based on slides from: Eric Bishop

Introduction: What is Unix?



- An operating system
- Developed at AT&T Bell Labs in the 1960's
- Command Line Interpreter
- GUIs (Window systems) are now available



Introduction: Unix vs. Linux

- Unix was the predecessor of Linux
- Linux is a variant of Unix
 - So is Mac OS X, so much of this tutorial applies to Macs as well
- Linux is open source

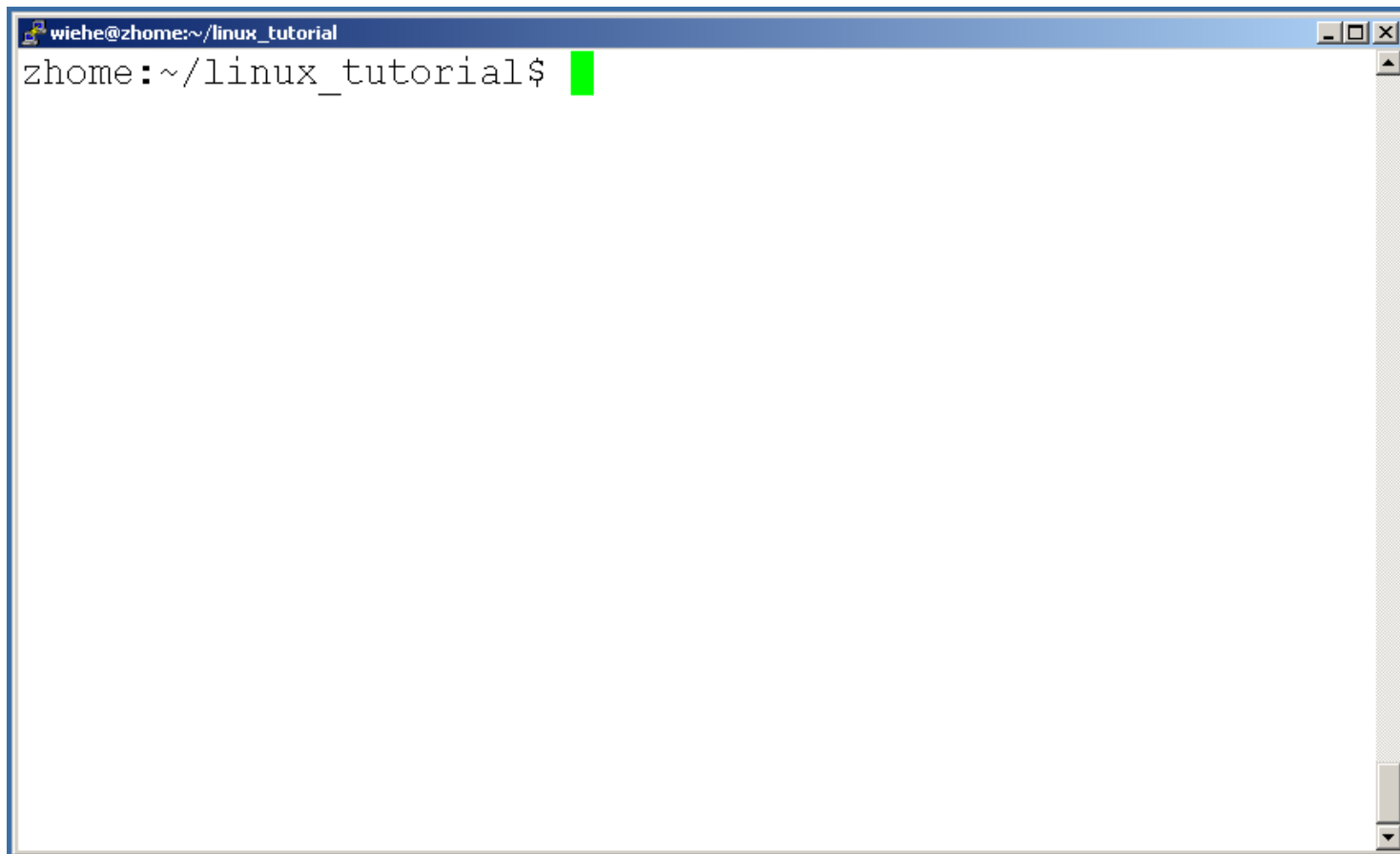
Introduction: Why Unix/Linux?



- Linux is **free**
- It's fully **customizable**
- It's **stable** (i.e. it almost never crashes)
- These characteristics make it an ideal OS for programmers and scientists

Connecting to a Unix/Linux system

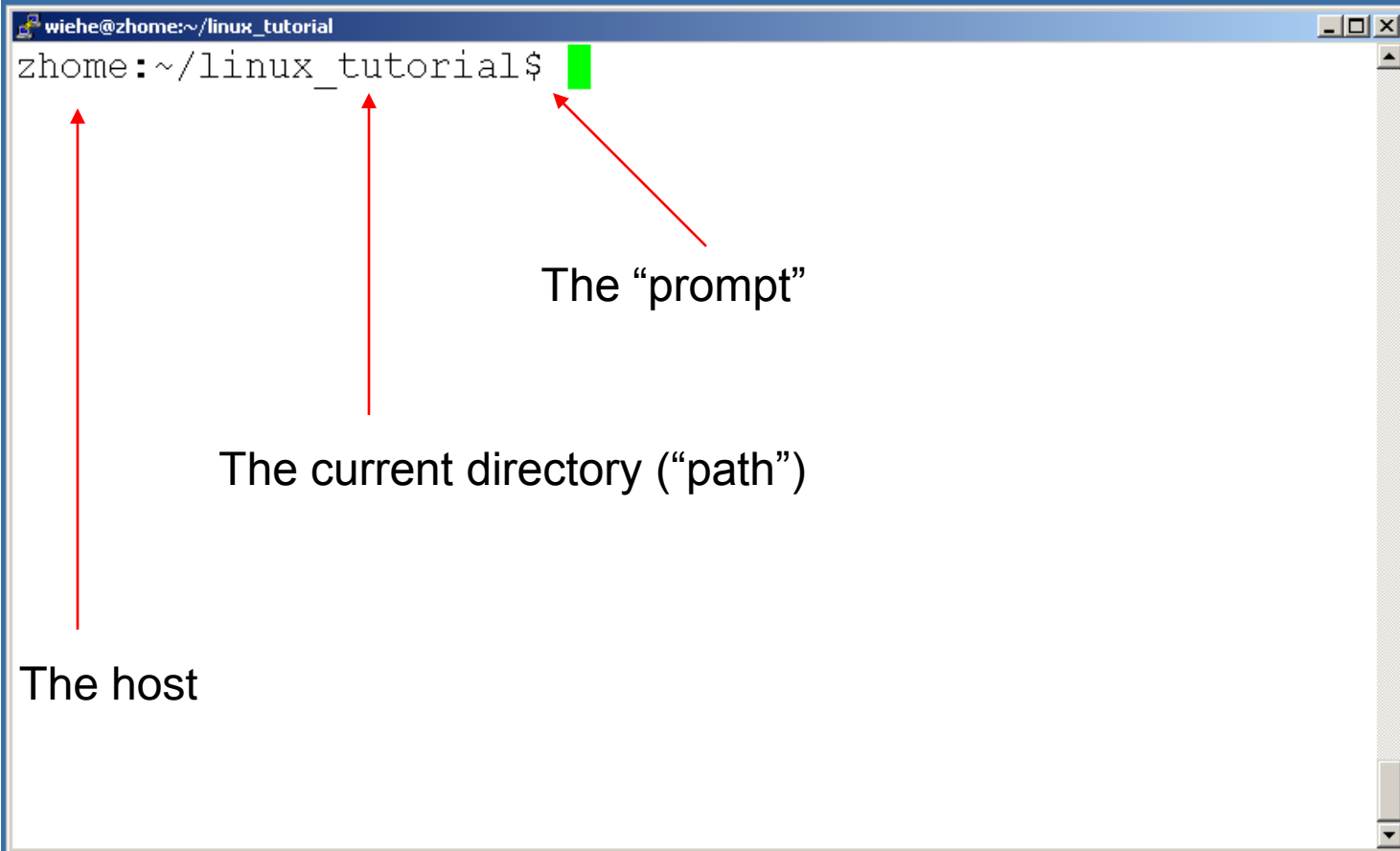
- Open up a terminal:

A screenshot of a terminal window. The title bar at the top reads "wiehe@zhome:~/linux_tutorial". The main area of the terminal shows the prompt "zhome:~/linux_tutorial\$" followed by a green cursor block. The window has standard Linux window controls (minimize, maximize, close) in the top right corner and a vertical scrollbar on the right side.

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ █
```

Connecting to a Unix/Linux system

- Open up a terminal:



The image shows a terminal window with the following text and annotations:

```
wiehe@zhome:~/linux_tutorial  
zhome:~/linux_tutorial$ █
```

Annotations with red arrows pointing to the prompt:

- The host (points to `wiehe@zhome`)
- The current directory ("path") (points to `~/linux_tutorial`)
- The "prompt" (points to `$`)



What exactly is a “shell”?

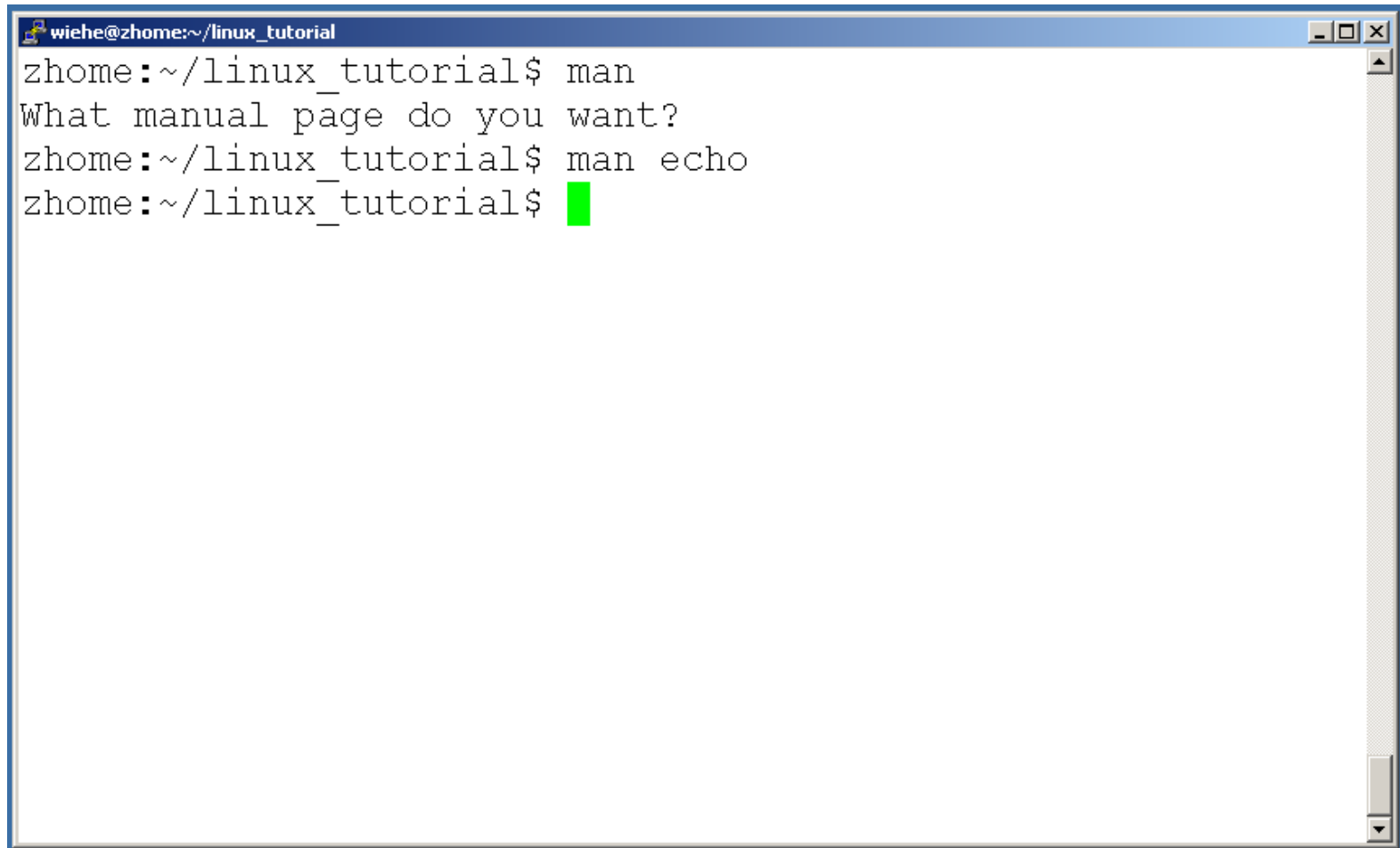
- After logging in, Linux/Unix starts another program called the **shell**
- The shell interprets commands the user types and manages their execution
 - The shell communicates with the internal part of the operating system called the **kernel**
 - The most popular shells are: tcsh, csh, korn, and bash
 - The differences are most times subtle
 - For this tutorial, we are using bash
- Shell commands are **CASE SENSITIVE!**



Help!

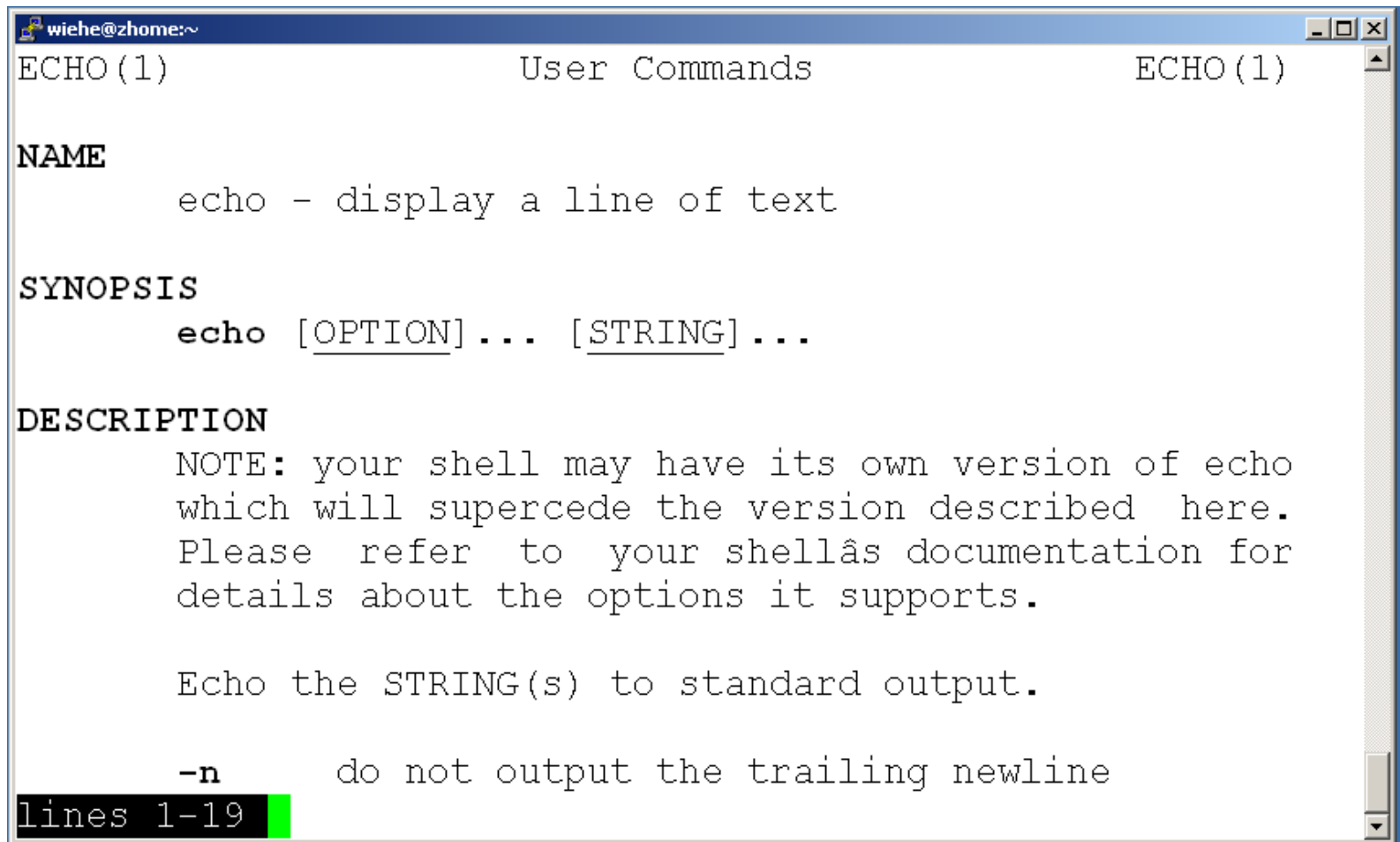
- Whenever you need help with a command type “man” and the command name

Help!



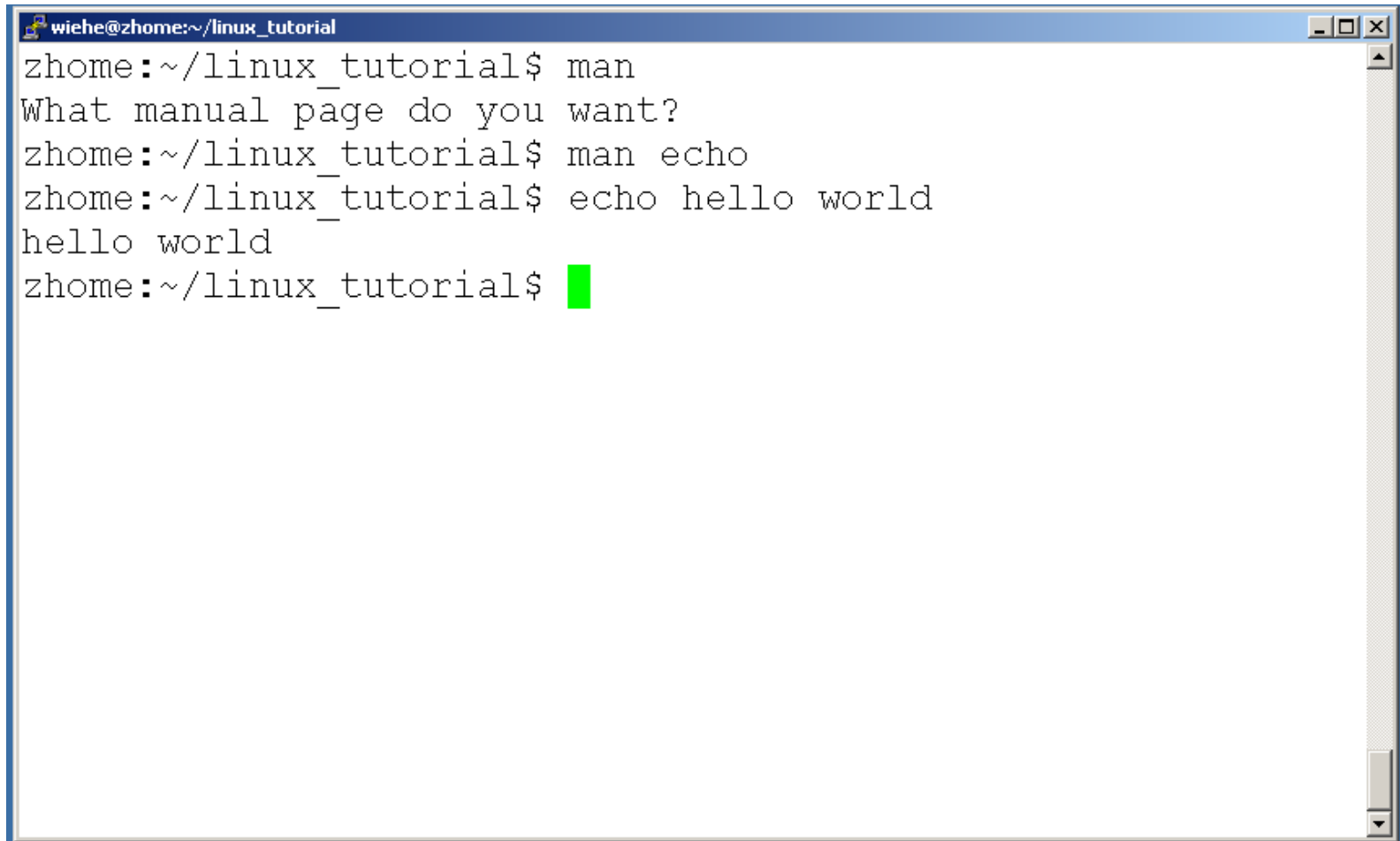
```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ man
What manual page do you want?
zhome:~/linux_tutorial$ man echo
zhome:~/linux_tutorial$ █
```

Help!



```
wiehe@zhome:~  
ECHO (1) User Commands ECHO (1)  
  
NAME  
    echo - display a line of text  
  
SYNOPSIS  
    echo [OPTION]... [STRING]...  
  
DESCRIPTION  
    NOTE: your shell may have its own version of echo  
    which will supercede the version described here.  
    Please refer to your shell's documentation for  
    details about the options it supports.  
  
    Echo the STRING(s) to standard output.  
  
    -n    do not output the trailing newline  
lines 1-19
```

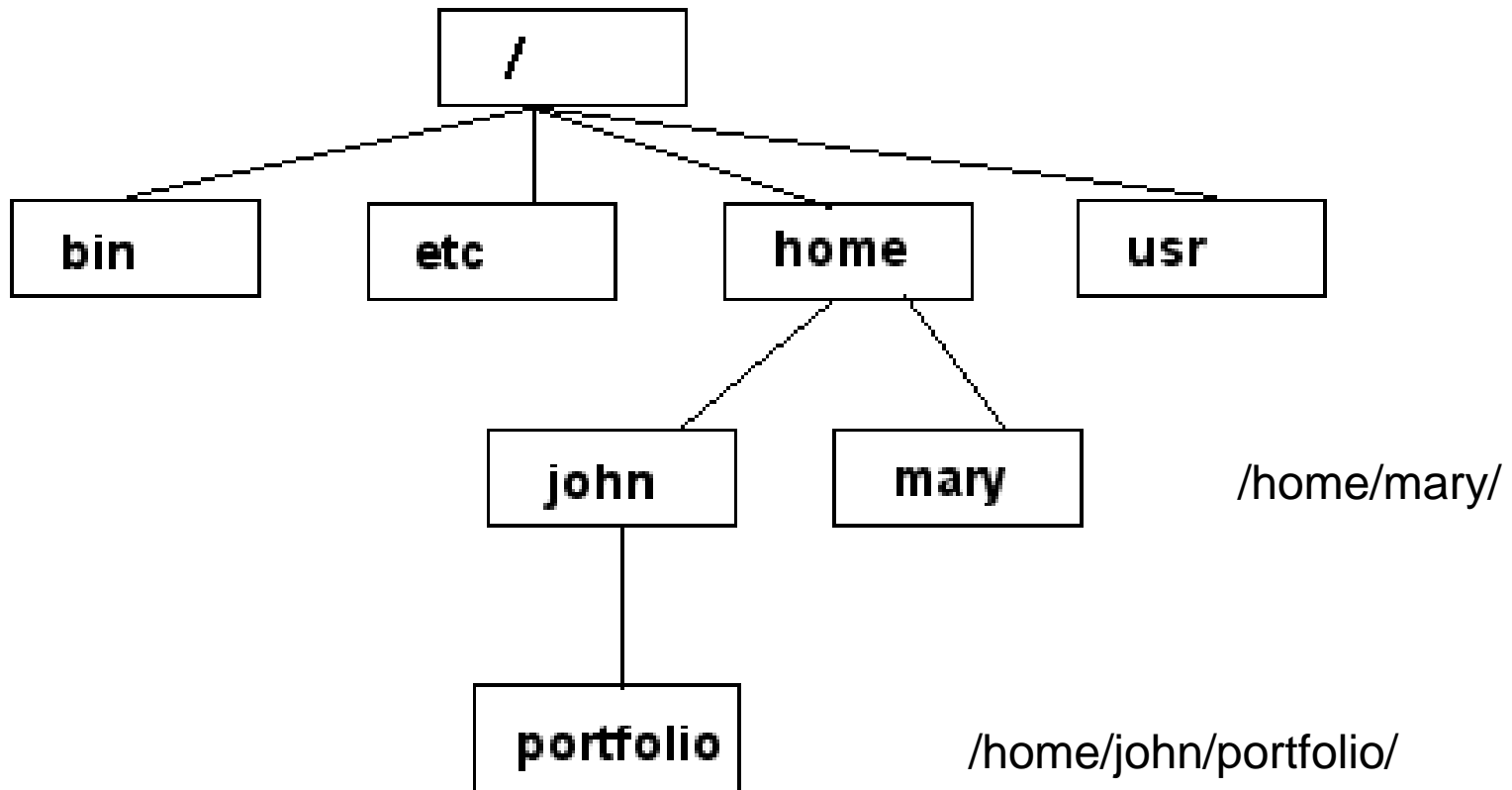
Help!



```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ man
What manual page do you want?
zhome:~/linux_tutorial$ man echo
zhome:~/linux_tutorial$ echo hello world
hello world
zhome:~/linux_tutorial$ █
```

Unix/Linux File System

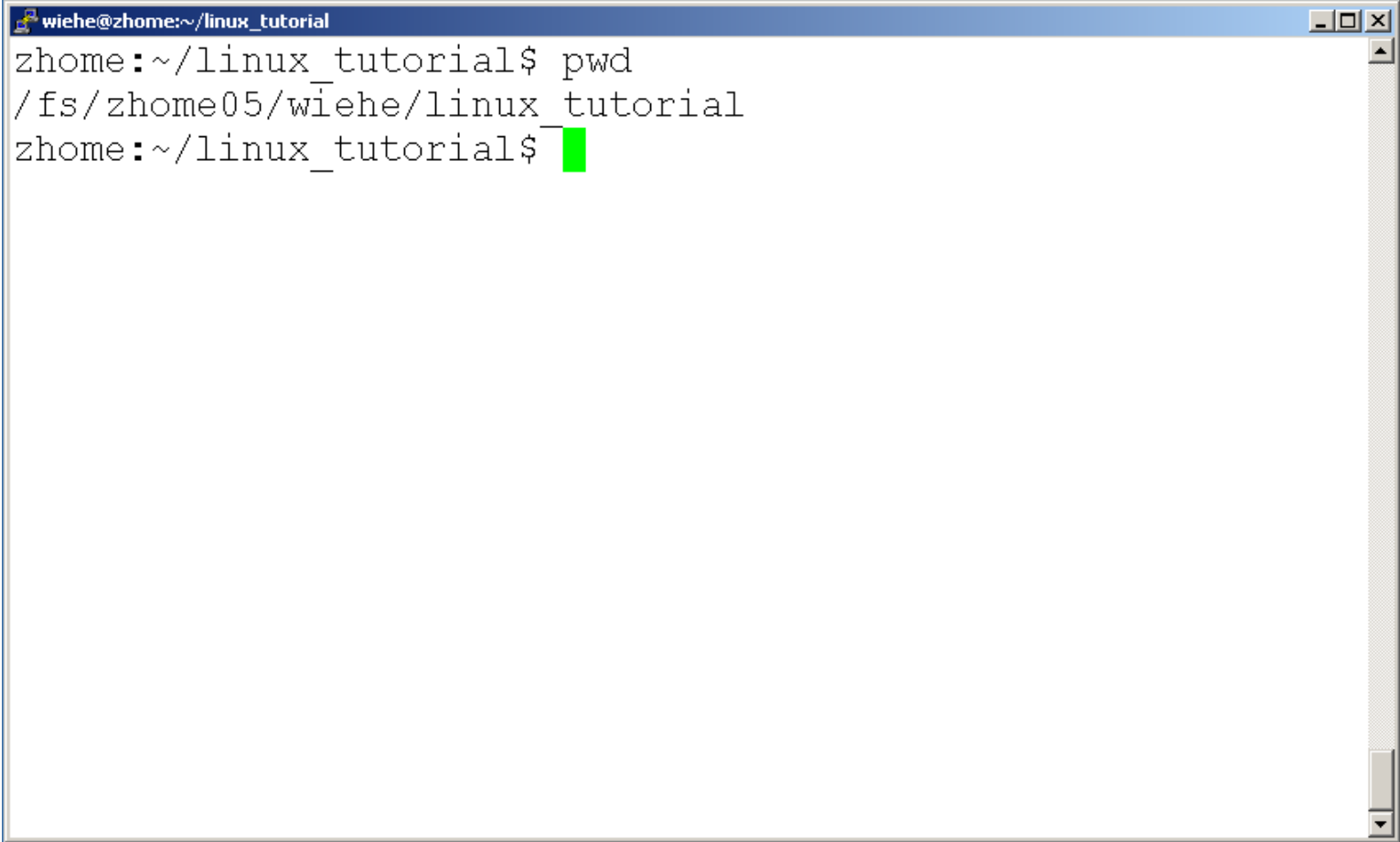
NOTE: Unix file names are **CASE SENSITIVE!**



The Path

Command: pwd

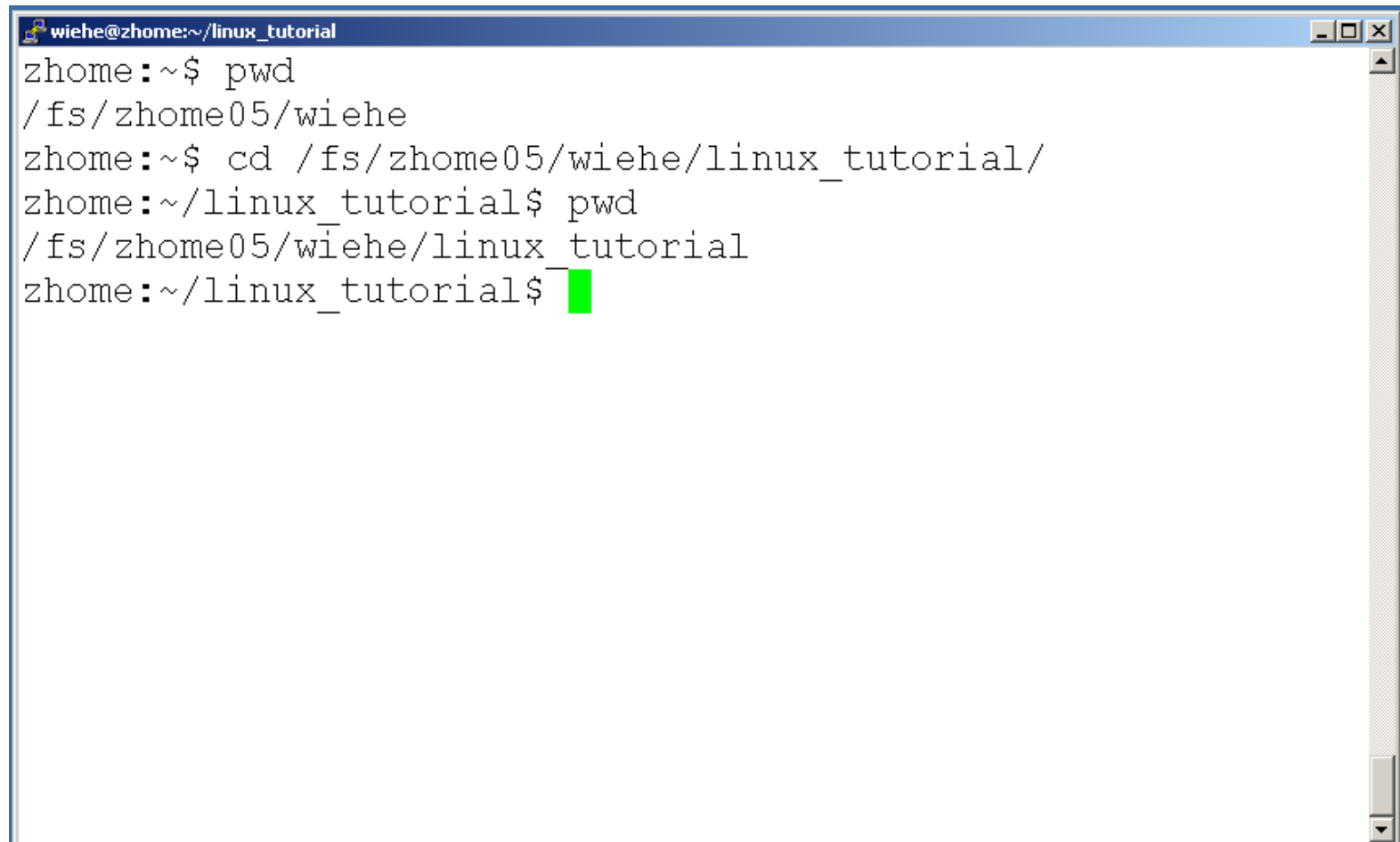
- To find your current path use “pwd”



```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ pwd
/fs/zhome05/wiehe/linux_tutorial
zhome:~/linux_tutorial$
```

Command: cd

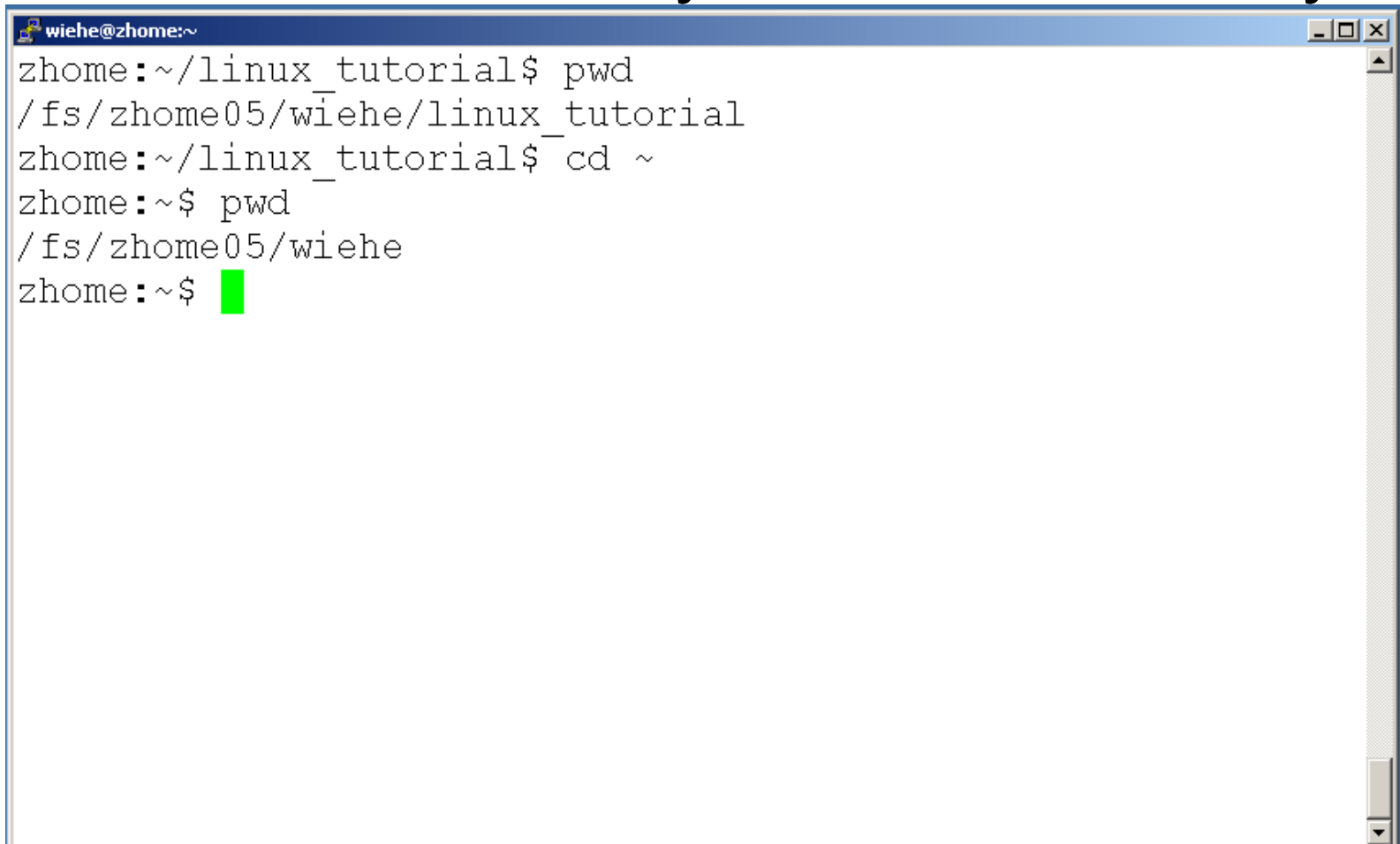
- To change to a specific directory use “cd”

A terminal window titled "wiehe@zhome:~/linux_tutorial" showing a sequence of commands and their outputs. The user starts in the home directory (~) and runs 'pwd', which outputs '/fs/zhome05/wiehe'. Then, the user runs 'cd /fs/zhome05/wiehe/linux_tutorial/' to change to the 'linux_tutorial' directory. Finally, the user runs 'pwd' again, which outputs '/fs/zhome05/wiehe/linux_tutorial'. The prompt is now 'zhome:~/linux_tutorial\$' with a green cursor.

```
wiehe@zhome:~/linux_tutorial
zhome:~$ pwd
/fs/zhome05/wiehe
zhome:~$ cd /fs/zhome05/wiehe/linux_tutorial/
zhome:~/linux_tutorial$ pwd
/fs/zhome05/wiehe/linux_tutorial
zhome:~/linux_tutorial$ █
```

Command: cd

- “~” is the location of your home directory



```
wiehe@zhome:~  
zhome:~/linux_tutorial$ pwd  
/fs/zhome05/wiehe/linux_tutorial  
zhome:~/linux_tutorial$ cd ~  
zhome:~$ pwd  
/fs/zhome05/wiehe  
zhome:~$ █
```

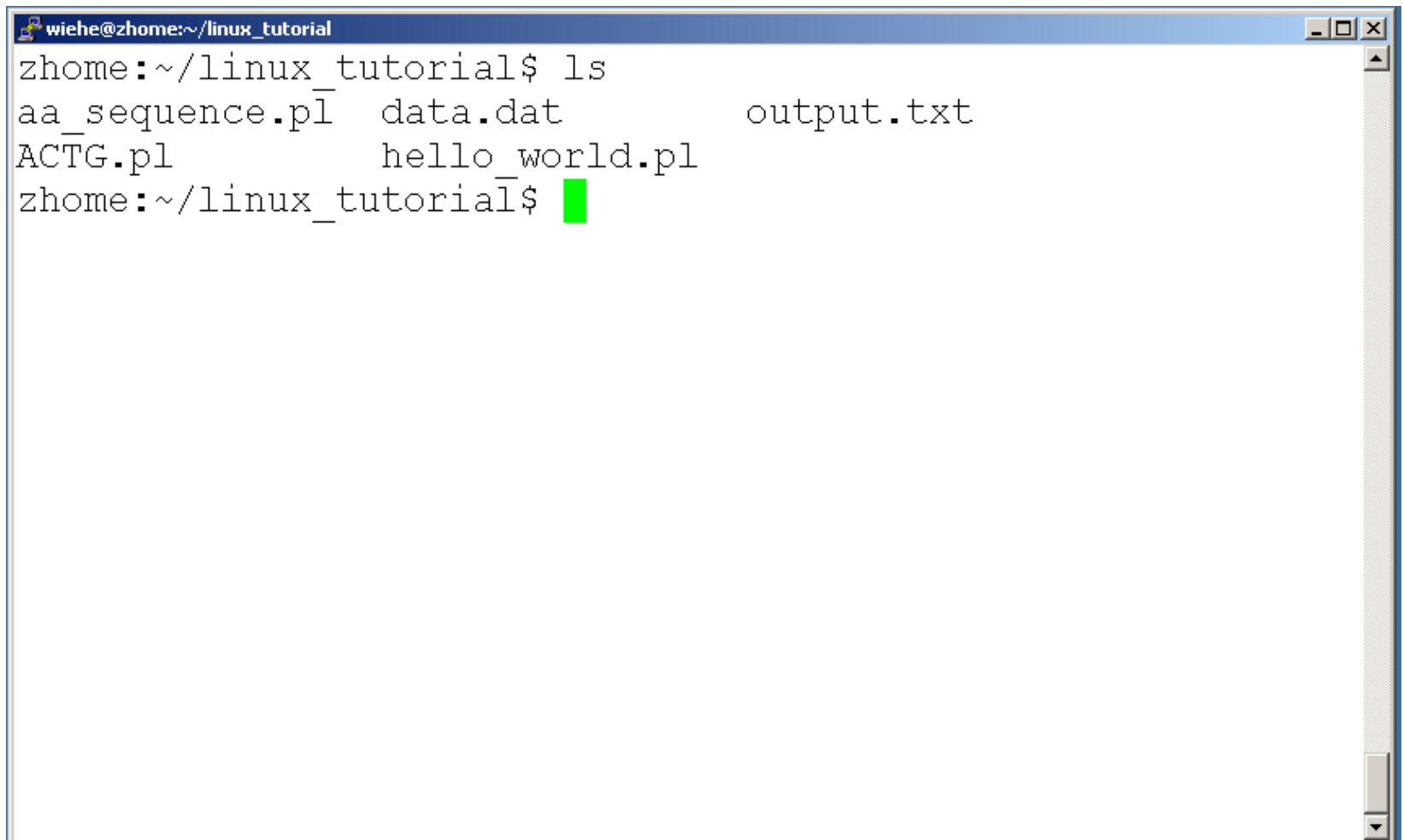
Command: cd

- “..” is the location of the directory below current one

```
wiehe@zhome:~  
zhome:~/linux_tutorial$ pwd  
/fs/zhome05/wiehe/linux_tutorial  
zhome:~/linux_tutorial$ cd ..  
zhome:~$ pwd  
/fs/zhome05/wiehe  
zhome:~$ █
```


Command: ls

- To list the files in the current directory use “ls”

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The terminal content shows the command 'ls' being executed, resulting in a list of files: 'aa_sequence.pl', 'data.dat', 'output.txt', and 'hello_world.pl'. The prompt 'zhome:~/linux_tutorial\$' is shown at the end of the output, followed by a green cursor.

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

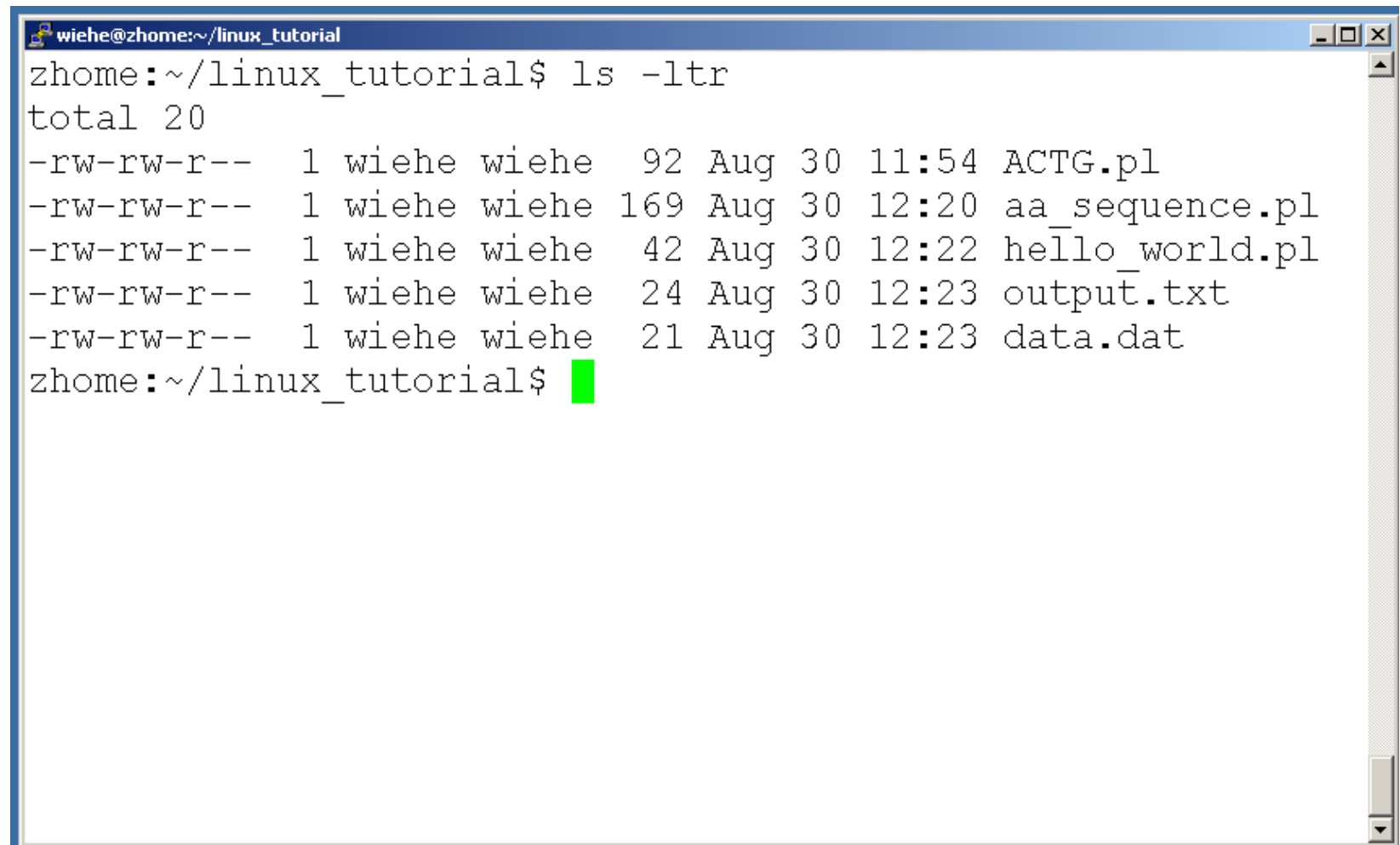


Command: ls

- ls has many options
 - -l long list (displays lots of info)
 - -t sort by modification time
 - -S sort by size
 - -h list file sizes in human readable format
 - -r reverse the order
- “man ls” for more options
- Options can be combined: “ls -ltr”

Command: ls -ltr

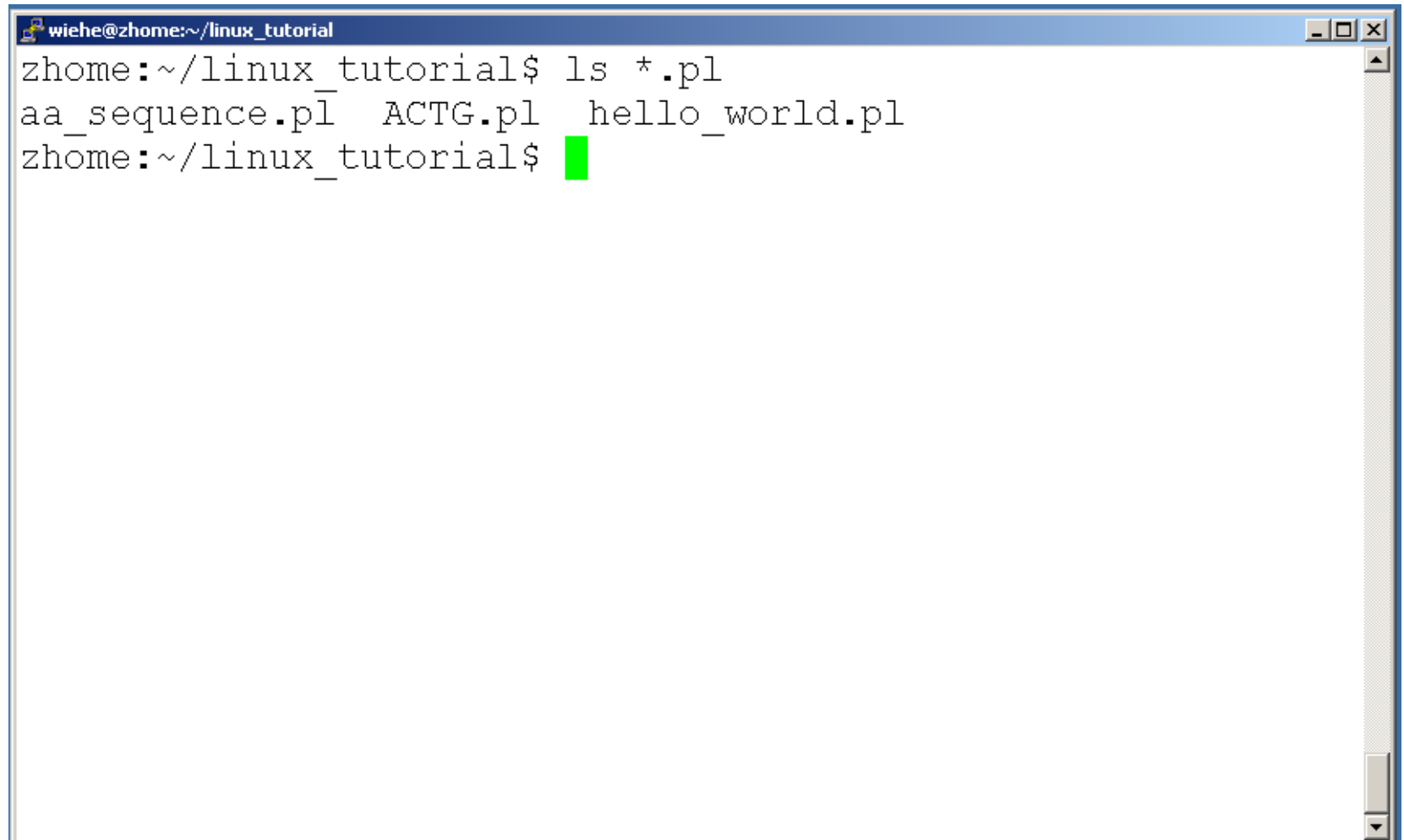
- List files by time in reverse order with long listing

A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing the output of the 'ls -ltr' command. The output lists five files in reverse chronological order: ACTG.pl (92 bytes, Aug 30 11:54), aa_sequence.pl (169 bytes, Aug 30 12:20), hello_world.pl (42 bytes, Aug 30 12:22), output.txt (24 bytes, Aug 30 12:23), and data.dat (21 bytes, Aug 30 12:23). The total size of the files is 20 bytes. The prompt 'zhome:~/linux_tutorial\$' is followed by a green cursor.

```
wiehe@zhome:~/linux_tutorial$ ls -ltr
total 20
-rw-rw-r--  1 wiehe wiehe  92 Aug 30 11:54 ACTG.pl
-rw-rw-r--  1 wiehe wiehe 169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r--  1 wiehe wiehe  42 Aug 30 12:22 hello_world.pl
-rw-rw-r--  1 wiehe wiehe  24 Aug 30 12:23 output.txt
-rw-rw-r--  1 wiehe wiehe  21 Aug 30 12:23 data.dat
zhome:~/linux_tutorial$
```

General Syntax: *

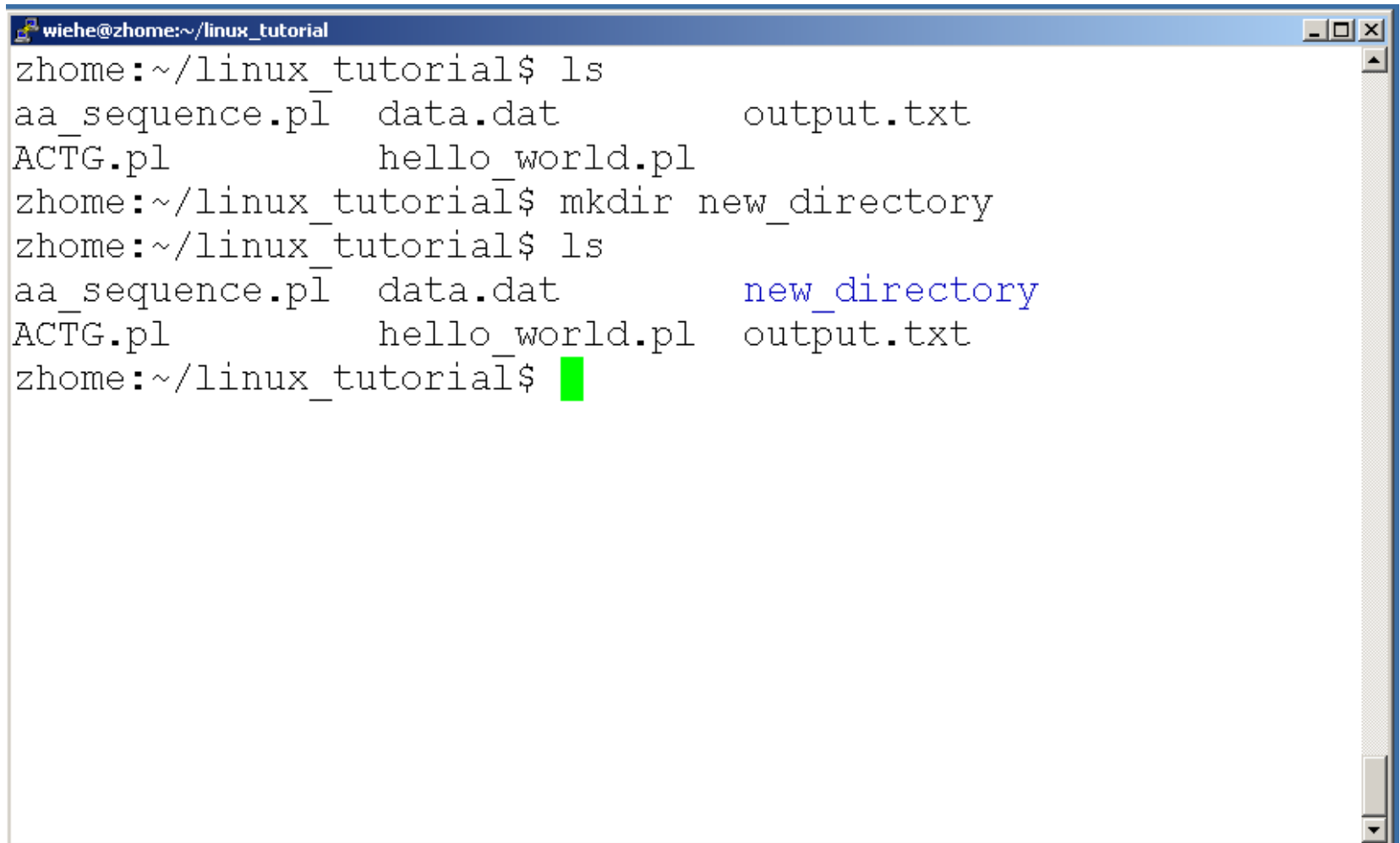
- “*” can be used as a wildcard in unix/linux

A terminal window with a blue title bar containing the text "wiehe@zhome:~/linux_tutorial". The terminal content shows a command prompt "zhome:~/linux_tutorial\$" followed by the command "ls *.pl". The output of the command is "aa_sequence.pl ACTG.pl hello_world.pl". Below the output, the prompt "zhome:~/linux_tutorial\$" is shown again with a green cursor block.

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls *.pl
aa_sequence.pl  ACTG.pl  hello_world.pl
zhome:~/linux_tutorial$
```

Command: mkdir

- To create a new directory use “mkdir”

A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing the execution of the 'mkdir' command. The window has a blue title bar and a scroll bar on the right. The text inside the terminal is as follows:

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

Command: rmdir

- To remove an empty directory use “rmdir”

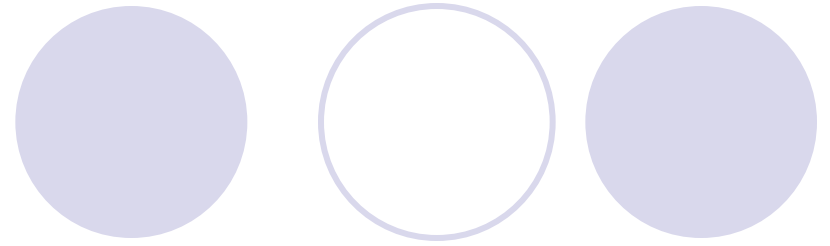
```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls
aa_sequence.pl  data.dat          new_directory
ACTG.pl        hello_world.pl   output.txt
zhome:~/linux_tutorial$ rmdir new_directory/
zhome:~/linux_tutorial$ ls
aa_sequence.pl  data.dat          output.txt
ACTG.pl        hello_world.pl
zhome:~/linux_tutorial$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

Working environments



- Writing C code:
 - IDE (Integrated development environment): emacs, eclipse, code::blocks, qtcreator etc.
 - Any text editor + SHELL commands.
 - Visual Studio – not recommended as does not support C99 and it is YOUR responsibility to migrate the code to linux.

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

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

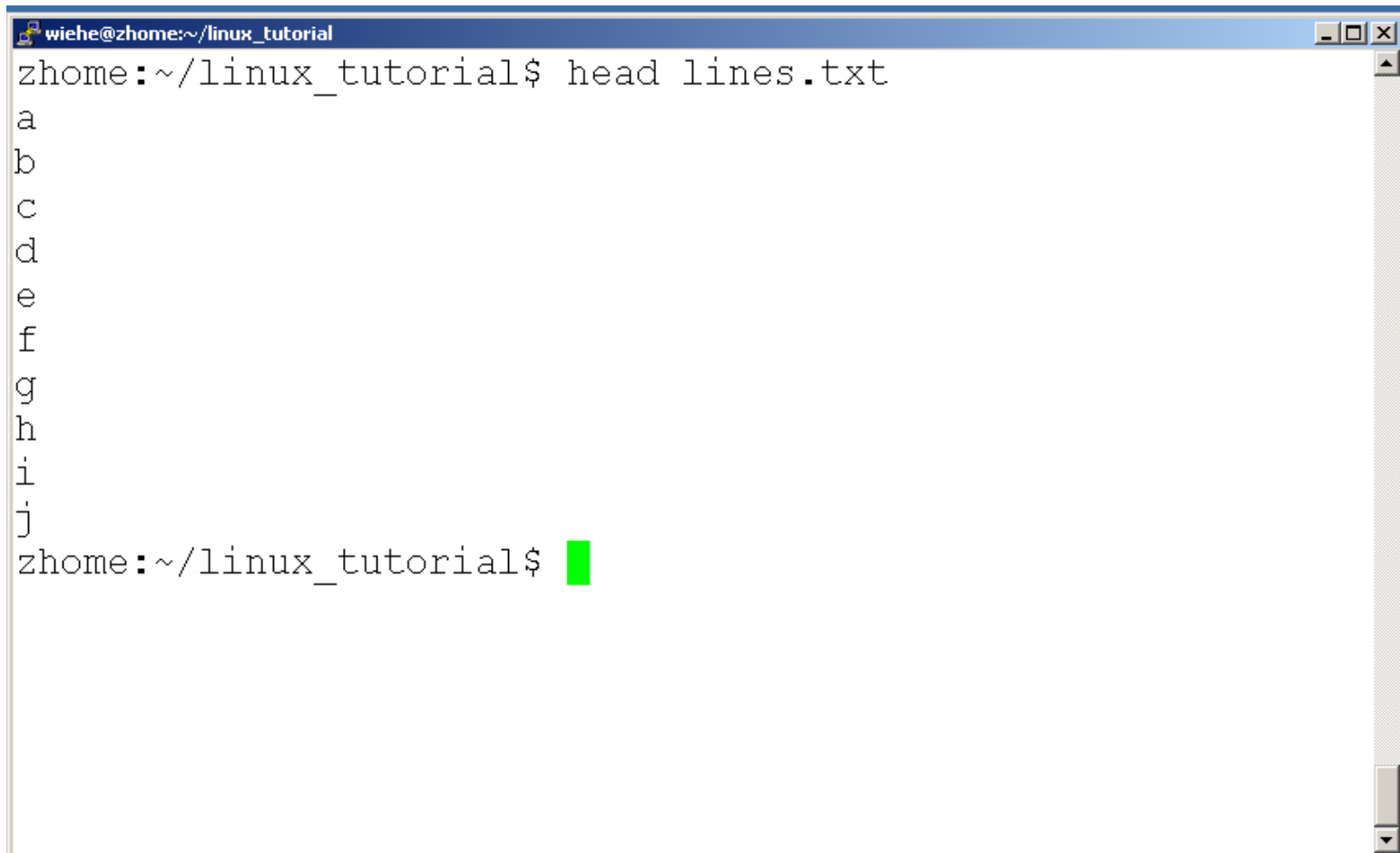
Command: head



- “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

Command: head

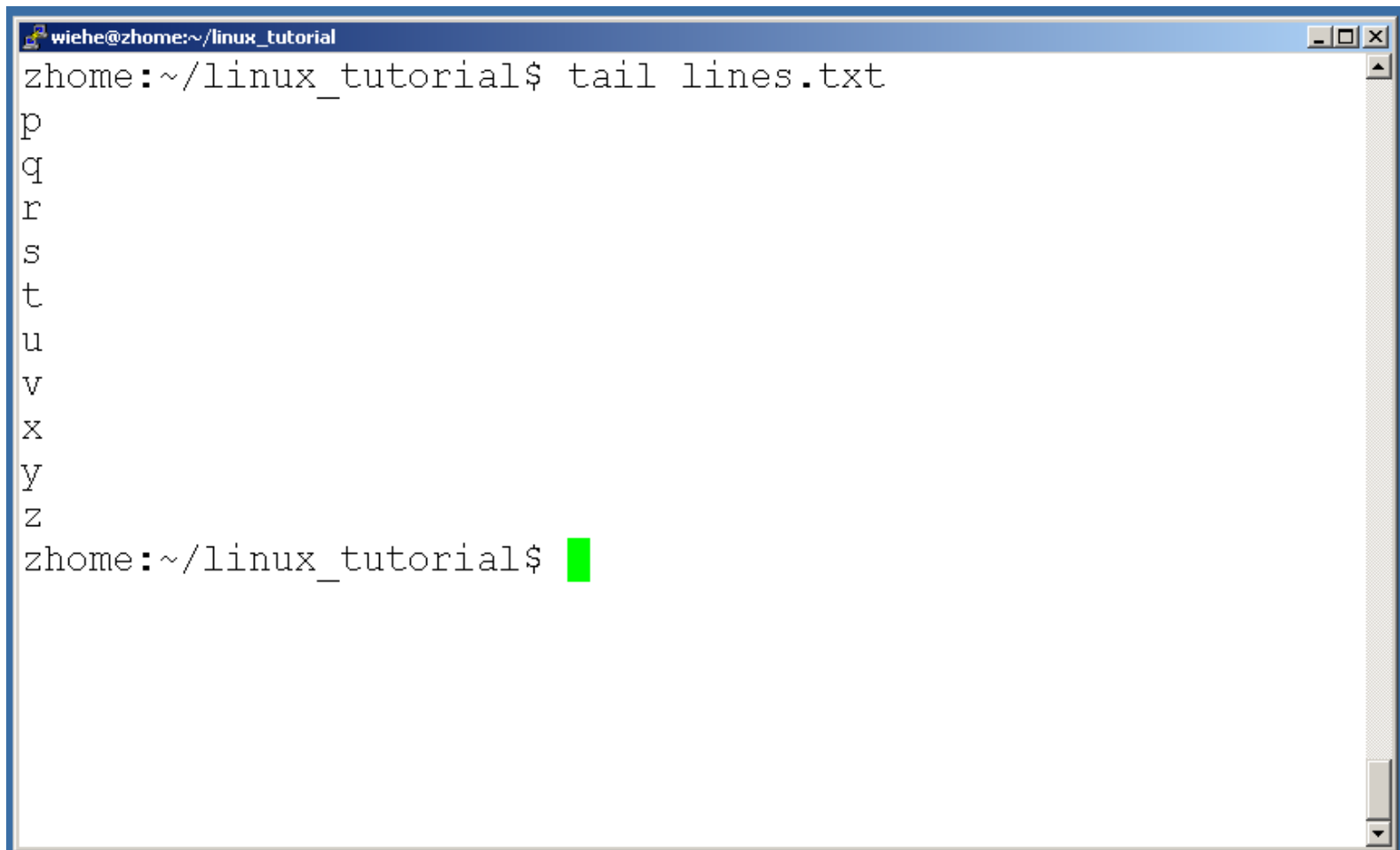
- Here's an example of using "head":

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The terminal shows the command 'head lines.txt' being executed, resulting in the output 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'. The prompt 'zhome:~/linux_tutorial\$' is visible at the end of the output, followed by a green cursor.

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

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 visible at the end of the output, 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$ █
```

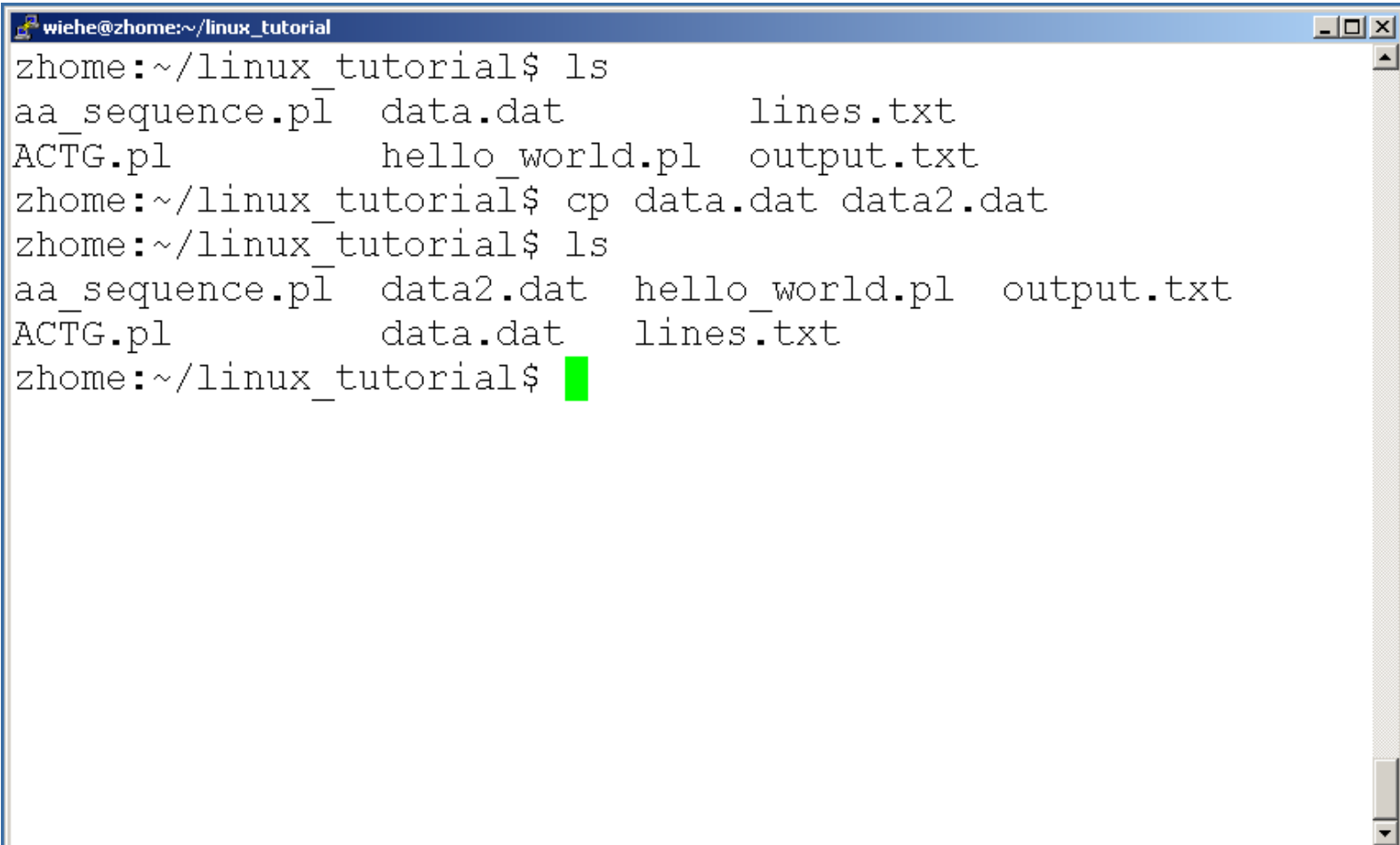
File Commands



- Copying a file: `cp`
- Move or rename a file: `mv`
- Remove a file: `rm`

Command: cp

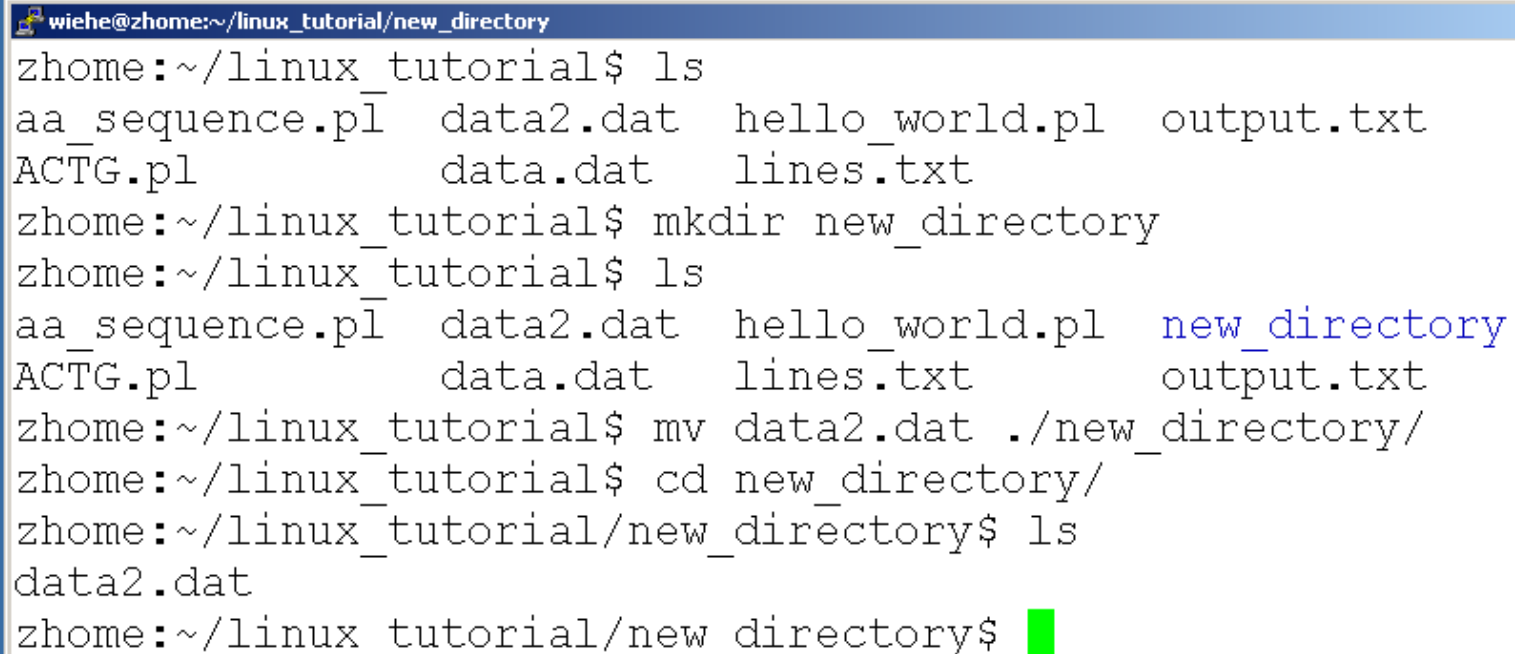
- To copy a file use “cp”



```
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$
```

Command: mv

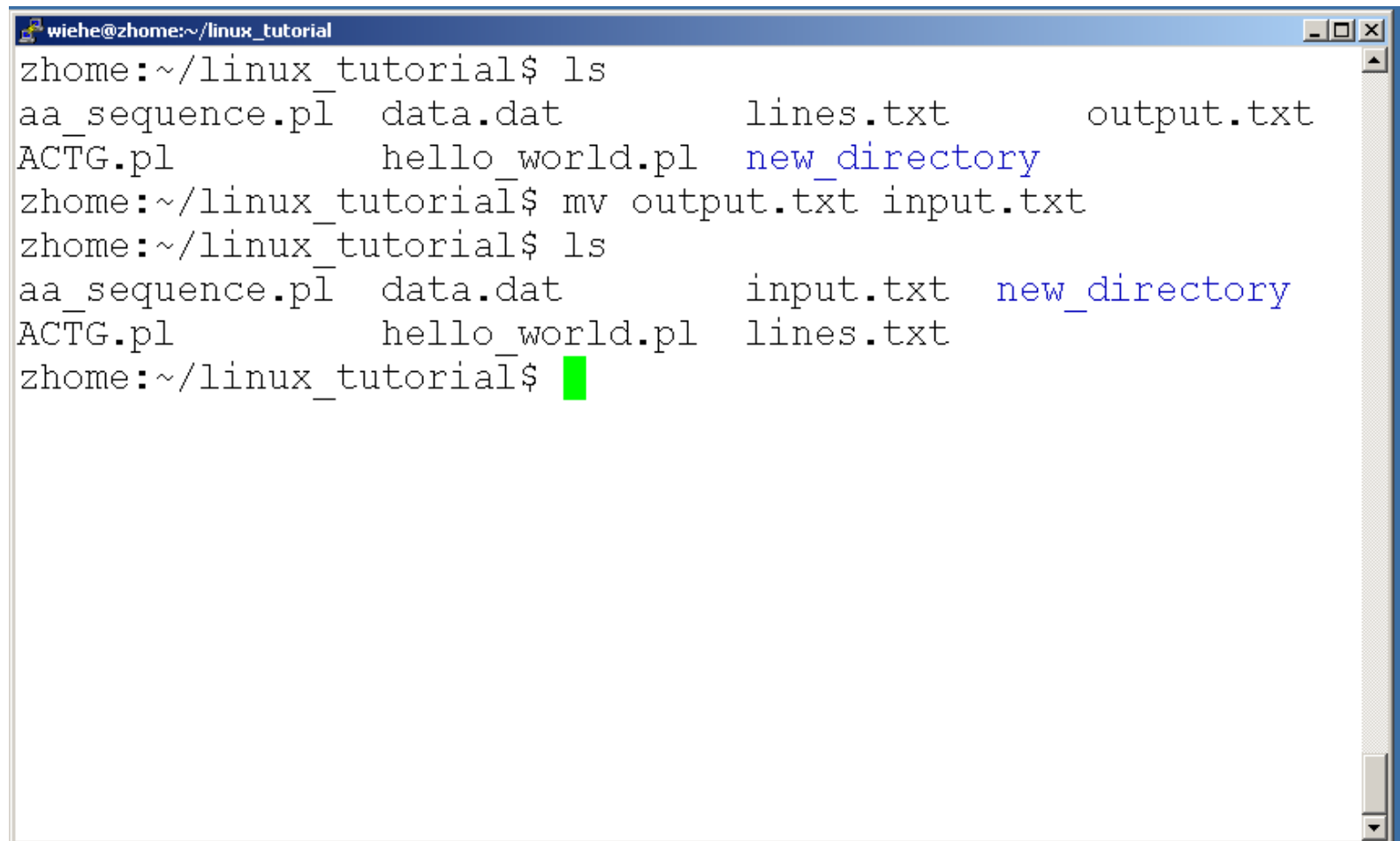
- To move a file to a different location use “mv”



```
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$
```


Command: mv

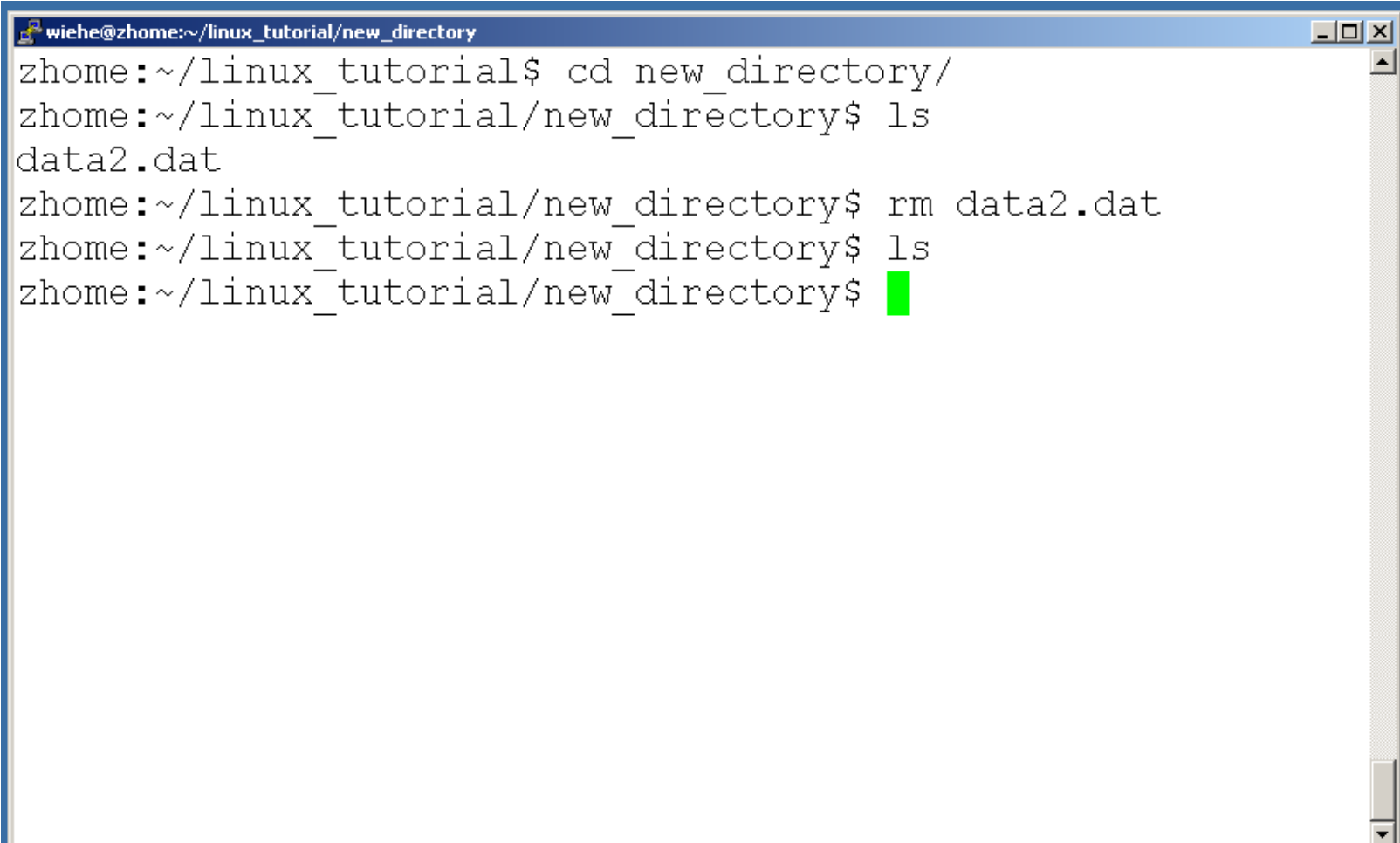
- mv can also be used to rename a file



```
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$ █
```

Command: rm

- To remove a file use “rm”



```
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$ █
```



Command: `rm`

- To remove a file “recursively”: `rm -r`
- Used to remove all files and directories
- Be very careful, deletions are permanent in Unix/Linux

File permissions



- Each file in Unix/Linux has an associated permission level
- This allows the user to prevent others from reading/writing/executing their files or directories
- Use “`ls -l filename`” to find the permission level of that file

Permission levels



- “r” means “read only” permission
- “w” means “write” permission
- “x” means “execute” permission
 - In case of directory, “x” grants permission to list directory contents

File Permissions

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls -l
total 28
-rw-rw-r-- 1 wiehe wiehe 169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r-- 1 wiehe wiehe 92 Aug 30 11:54 ACTG.pl
-rw-rw-r-- 1 wiehe wiehe 21 Aug 30 12:23 data.dat
-rw-rw-r-- 1 wiehe wiehe 42 Aug 30 12:22 hello_world.pl
-rw-rw-r-- 1 wiehe wiehe 24 Aug 30 12:23 input.txt
-rw-rw-r-- 1 wiehe wiehe 50 Aug 30 13:13 lines.txt
drwxrwxr-x 2 wiehe wiehe 4096 Aug 30 13:19 new_directory
zhome:~/linux_tutorial$
```

User (you)

File Permissions

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls -l
total 28
-rw-rw-r--  1 wiehe wiehe  169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r--  1 wiehe wiehe   92 Aug 30 11:54 ACTG.pl
-rw-rw-r--  1 wiehe wiehe   21 Aug 30 12:23 data.dat
-rw-rw-r--  1 wiehe wiehe   42 Aug 30 12:22 hello_world.pl
-rw-rw-r--  1 wiehe wiehe   24 Aug 30 12:23 input.txt
-rw-rw-r--  1 wiehe wiehe   50 Aug 30 13:13 lines.txt
drwxrwxr-x  2 wiehe wiehe 4096 Aug 30 13:19 new_directory
zhome:~/linux_tutorial$
```

Group

File Permissions

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls -l
total 28
-rw-rw-r-- 1 wiehe wiehe 169 Aug 30 12:20 aa_sequence.pl
-rw-rw-r-- 1 wiehe wiehe 92 Aug 30 11:54 ACTG.pl
-rw-rw-r-- 1 wiehe wiehe 21 Aug 30 12:23 data.dat
-rw-rw-r-- 1 wiehe wiehe 42 Aug 30 12:22 hello_world.pl
-rw-rw-r-- 1 wiehe wiehe 24 Aug 30 12:23 input.txt
-rw-rw-r-- 1 wiehe wiehe 50 Aug 30 13:13 lines.txt
drwxrwxr-x 2 wiehe wiehe 4096 Aug 30 13:19 new_directory
zhome:~/linux_tutorial$
```

“The World”

Command: chmod

- If you own the file, you can change it's permissions with “chmod”
 - Syntax: chmod [**u**ser/**g**roup/**o**thers/**a**ll]+[permission] [file(s)]
 - Below we grant execute permission to all:

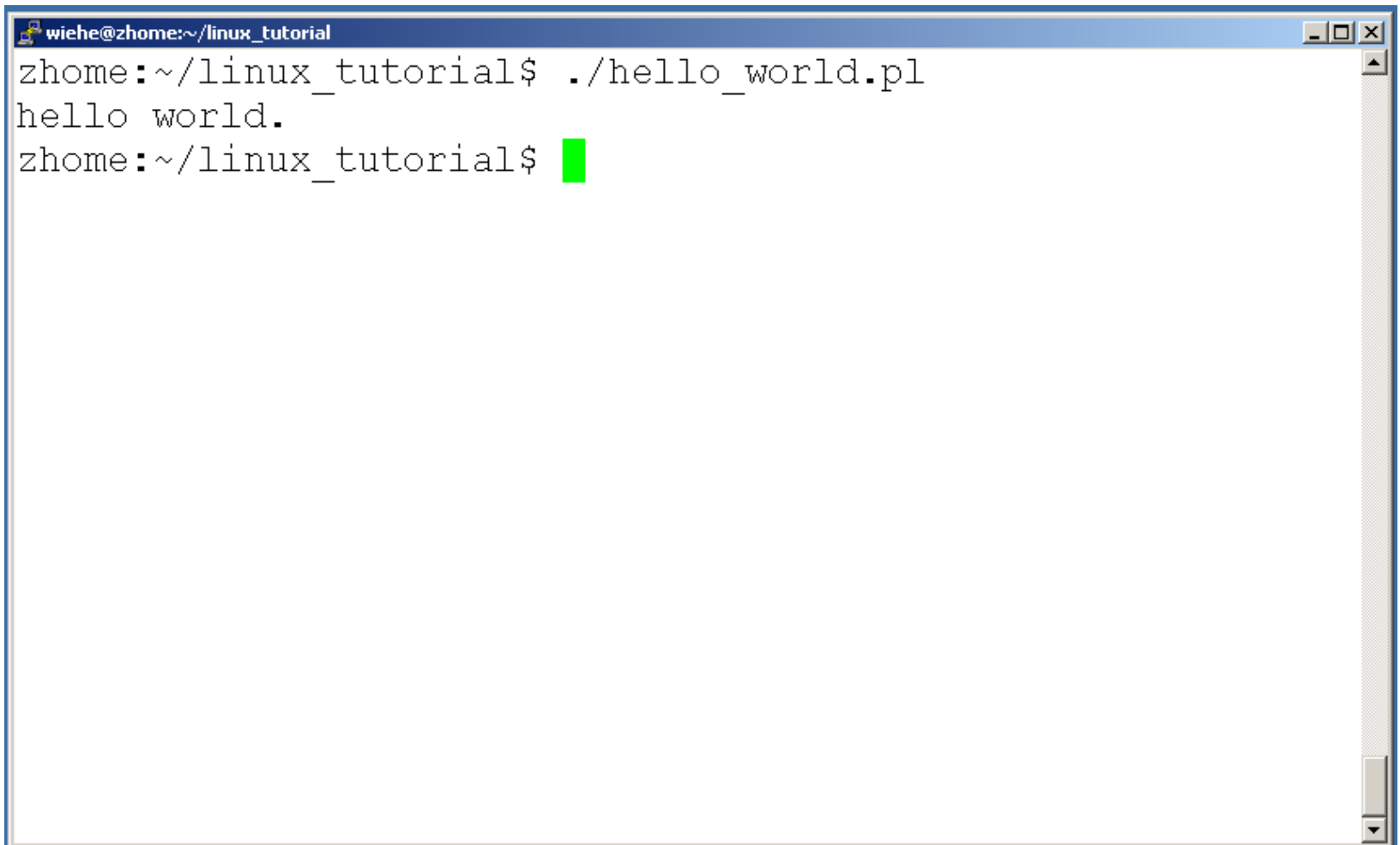
```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls -l hello_world.pl
-rw-rw-r-- 1 wiehe wiehe 42 Aug 30 12:22 hello_world.pl
zhome:~/linux_tutorial$ chmod a+x hello_world.pl
zhome:~/linux_tutorial$ ls -l hello_world.pl
-rwxrwxr-x 1 wiehe wiehe 42 Aug 30 12:22 hello_world.pl
zhome:~/linux_tutorial$
```

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

- Make sure the program has executable permissions
- Use “./” to run the program

Running a program: an example

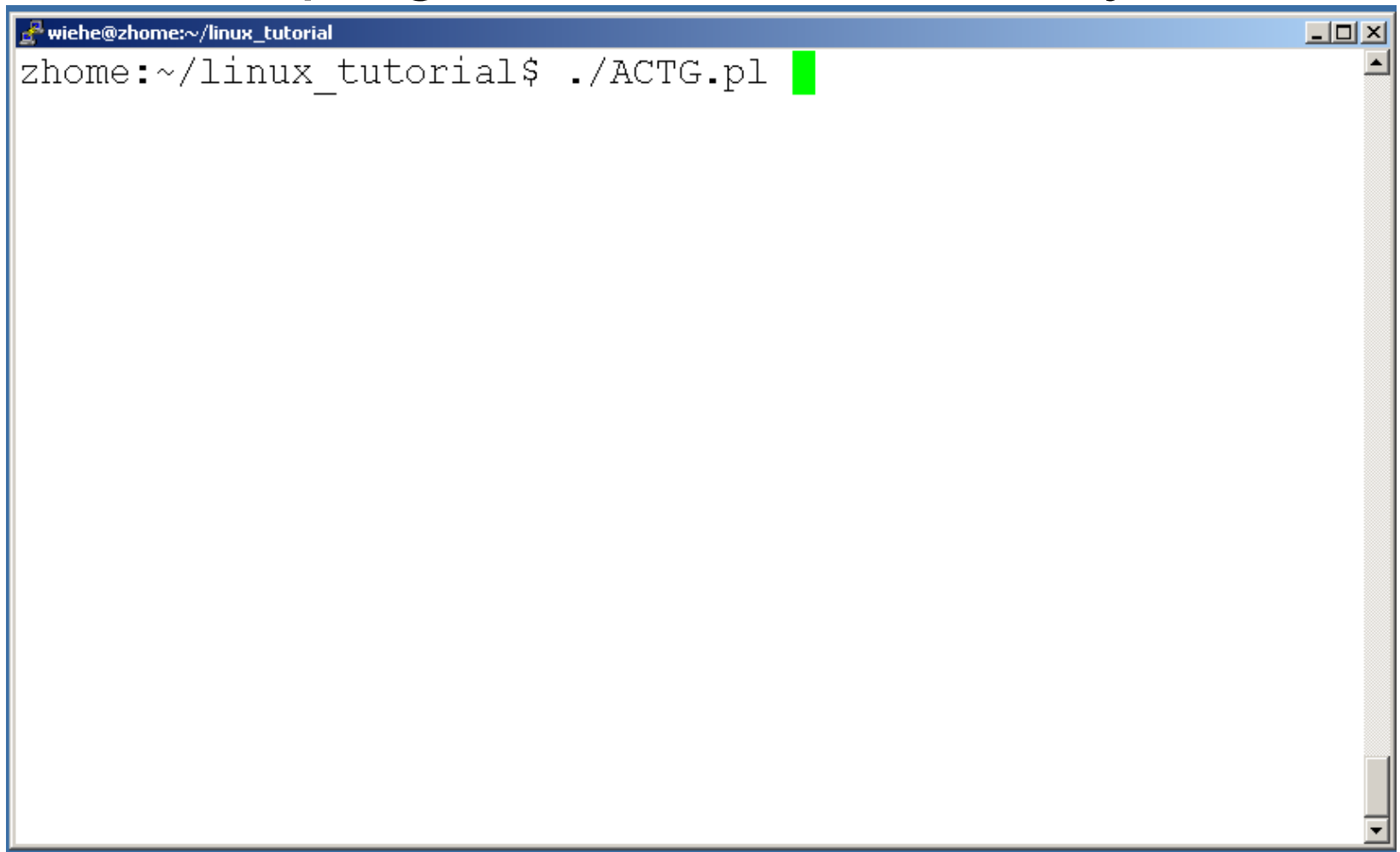
- Running the sample perl script “hello_world.pl”

A terminal window with a blue title bar containing the text "wiehe@zhome:~/linux_tutorial". The terminal content shows a user prompt "zhome:~/linux_tutorial\$" followed by the command "./hello_world.pl". The output is "hello world." on the next line. A second prompt "zhome:~/linux_tutorial\$" is shown with a green cursor block. The window has standard Linux window controls (minimize, maximize, close) in the top right and a scrollbar on the right side.

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

Ending a program

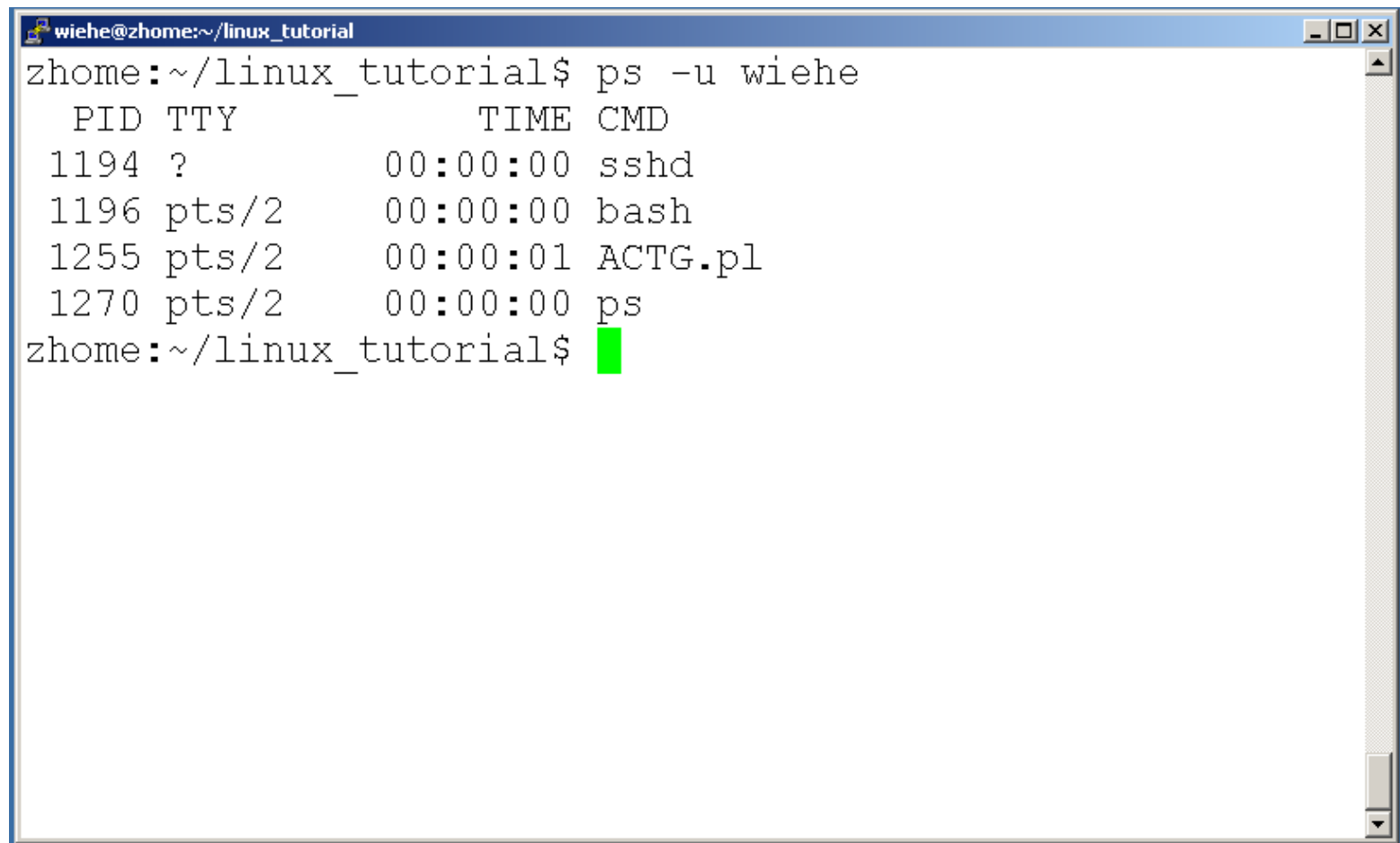
- To end a program use “ctrl-c”. To try it:

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The terminal content shows the prompt 'zhome:~/linux_tutorial\$' followed by the command './ACTG.pl' and a green cursor block. The window has standard Linux window controls (minimize, maximize, close) in the top right and a vertical scrollbar on the right side.

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ./ACTG.pl
```

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:

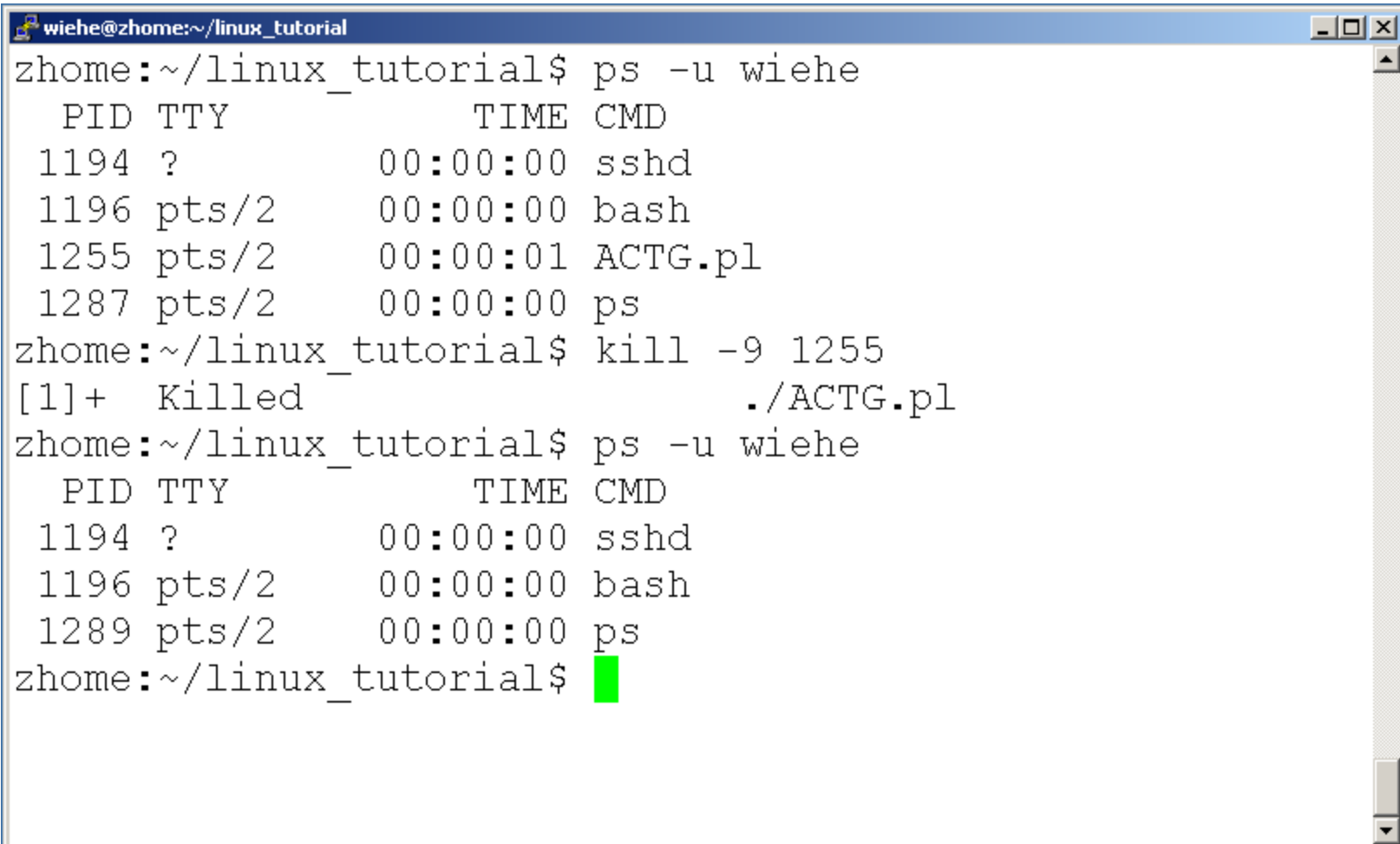
```
wiehe@zhome:~/linux_tutorial
top - 13:46:33 up 50 days,  4:26,  2 users,  load avera
Tasks:  total,      running,      sleeping,      stoppe
Cpu(s) :    us,      sy,      ni,      id,      w
Mem:      total,      used,      free,
Swap:      total,      used,      free,

```

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”

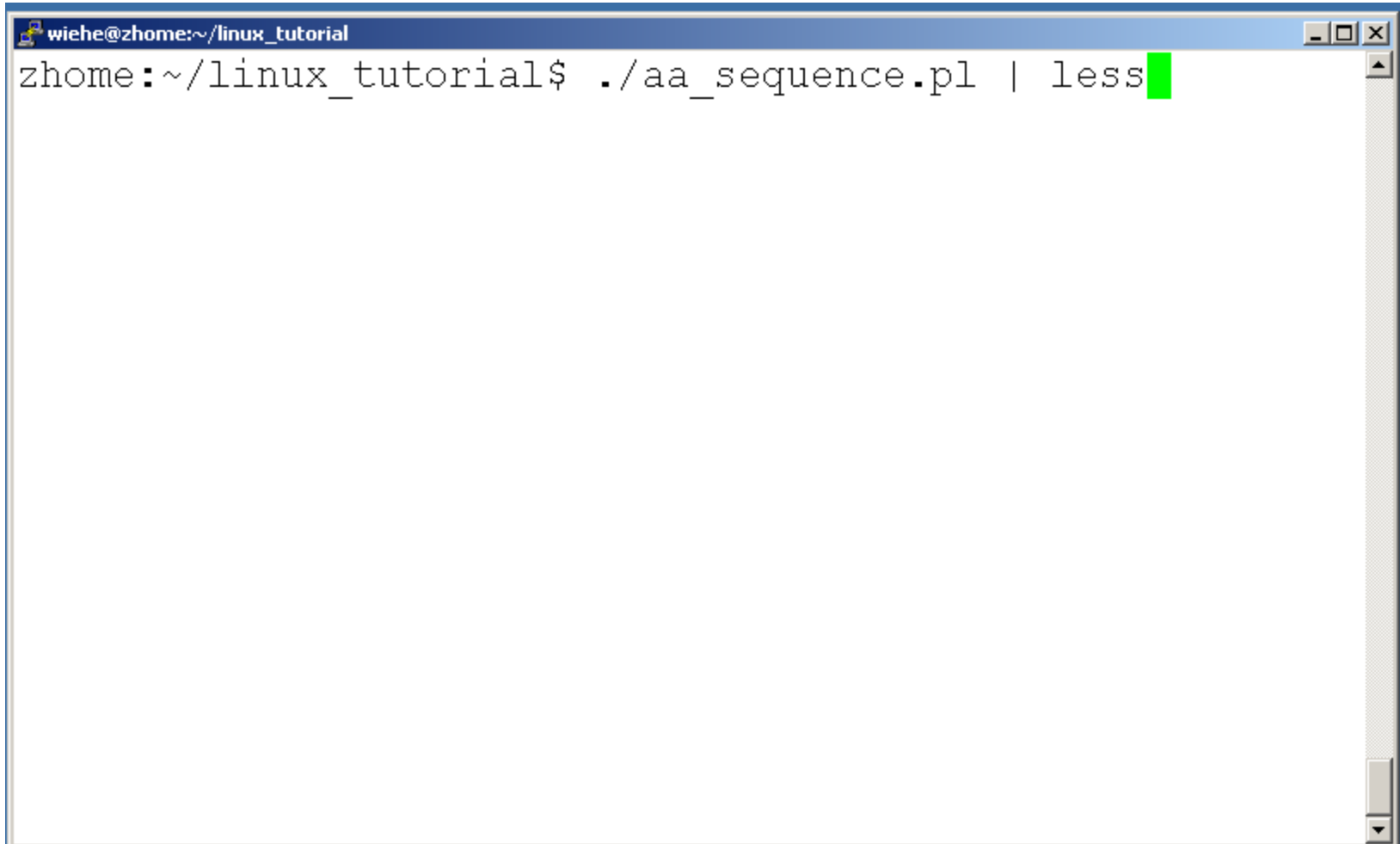


```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ps -u wiehe
  PID TTY          TIME CMD
 1194 ?            00:00:00 sshd
 1196 pts/2        00:00:00 bash
 1255 pts/2        00:00:01 ACTG.pl
 1287 pts/2        00:00:00 ps
zhome:~/linux_tutorial$ kill -9 1255
[1]+  Killed                  ./ACTG.pl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

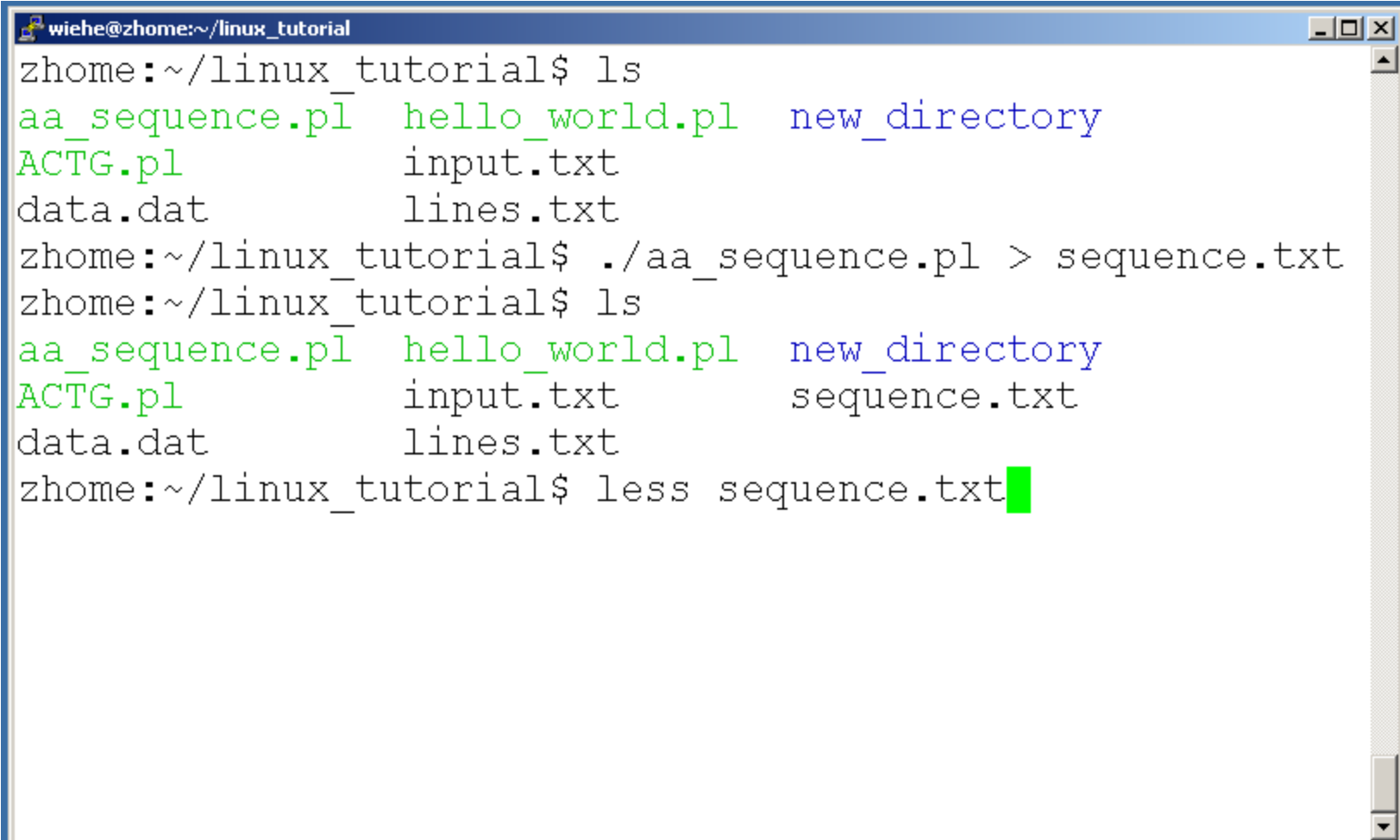
A few examples of piping



```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ./aa_sequence.pl | less
```

The image shows a terminal window with a blue title bar. The title bar contains the text 'wiehe@zhome:~/linux_tutorial' and standard window control icons (minimize, maximize, close). The terminal content shows a prompt 'zhome:~/linux_tutorial\$' followed by the command './aa_sequence.pl | less'. A green cursor is positioned at the end of the command. The terminal area is mostly empty, suggesting the output of the command is being viewed in a pager.

A few examples of piping



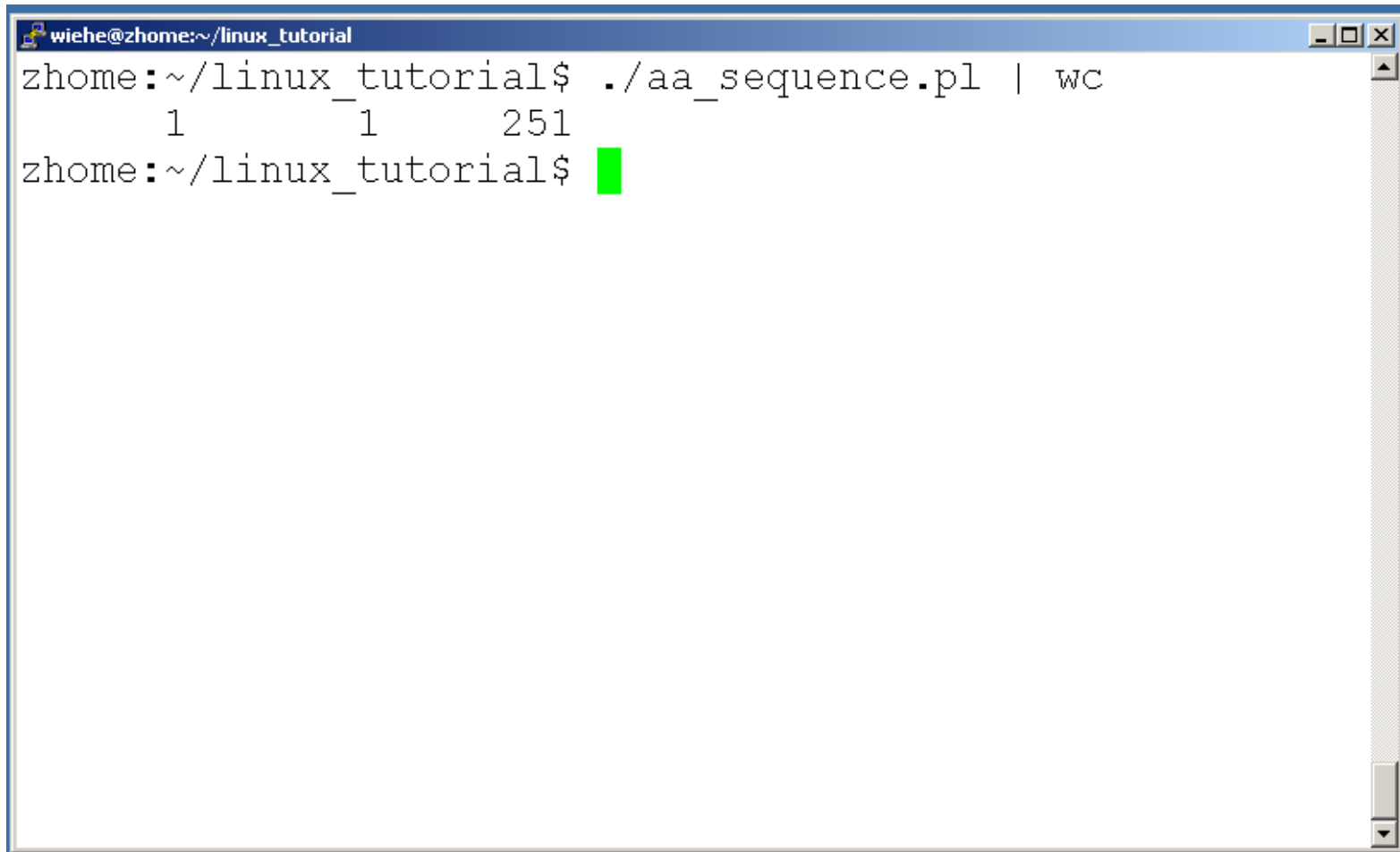
```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls
aa_sequence.pl  hello_world.pl  new_directory
ACTG.pl        input.txt
data.dat       lines.txt
zhome:~/linux_tutorial$ ./aa_sequence.pl > sequence.txt
zhome:~/linux_tutorial$ ls
aa_sequence.pl  hello_world.pl  new_directory
ACTG.pl        input.txt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

A few examples of piping

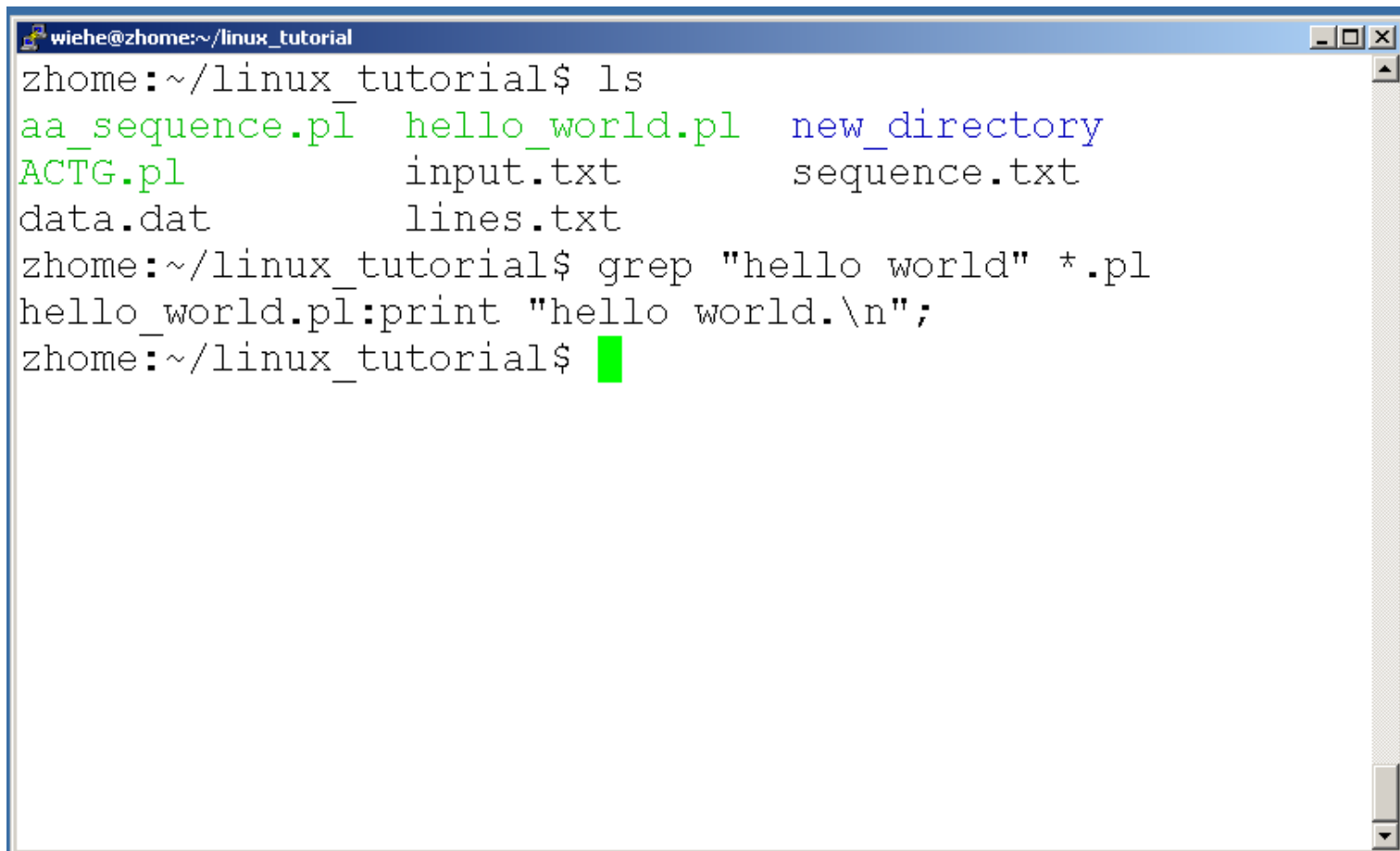


```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ./aa_sequence.pl | wc
      1      1    251
zhome:~/linux_tutorial$
```

The image shows a terminal window with a blue title bar containing the text "wiehe@zhome:~/linux_tutorial". The terminal content shows a command being executed: `./aa_sequence.pl | wc`. The output of this command is a single line of text: `1 1 251`. Below the output, the prompt `zhome:~/linux_tutorial$` is followed by a green cursor block, indicating the terminal is ready for the next command.

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

Unix Web Resources



- <http://www.ee.surrey.ac.uk/Teaching/Unix/>
- <http://www.ugu.com/sui/ugu/show?help.beginners>
- <http://en.wikipedia.org/wiki/Unix>

A decorative header consisting of five circles in a row. From left to right: a solid light purple circle, an empty light purple circle outline, a solid light purple circle, an empty light purple circle outline, and a solid light purple circle.

Emacs reference card

- <http://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf>