

# Introduction to Linux OS and Commands

# Introduction: What is Unix?

- An operating system
- Developed at AT&T Bell Labs in the 1960' s
- Command Line Interpreter

# Introduction: Unix vs. Linux

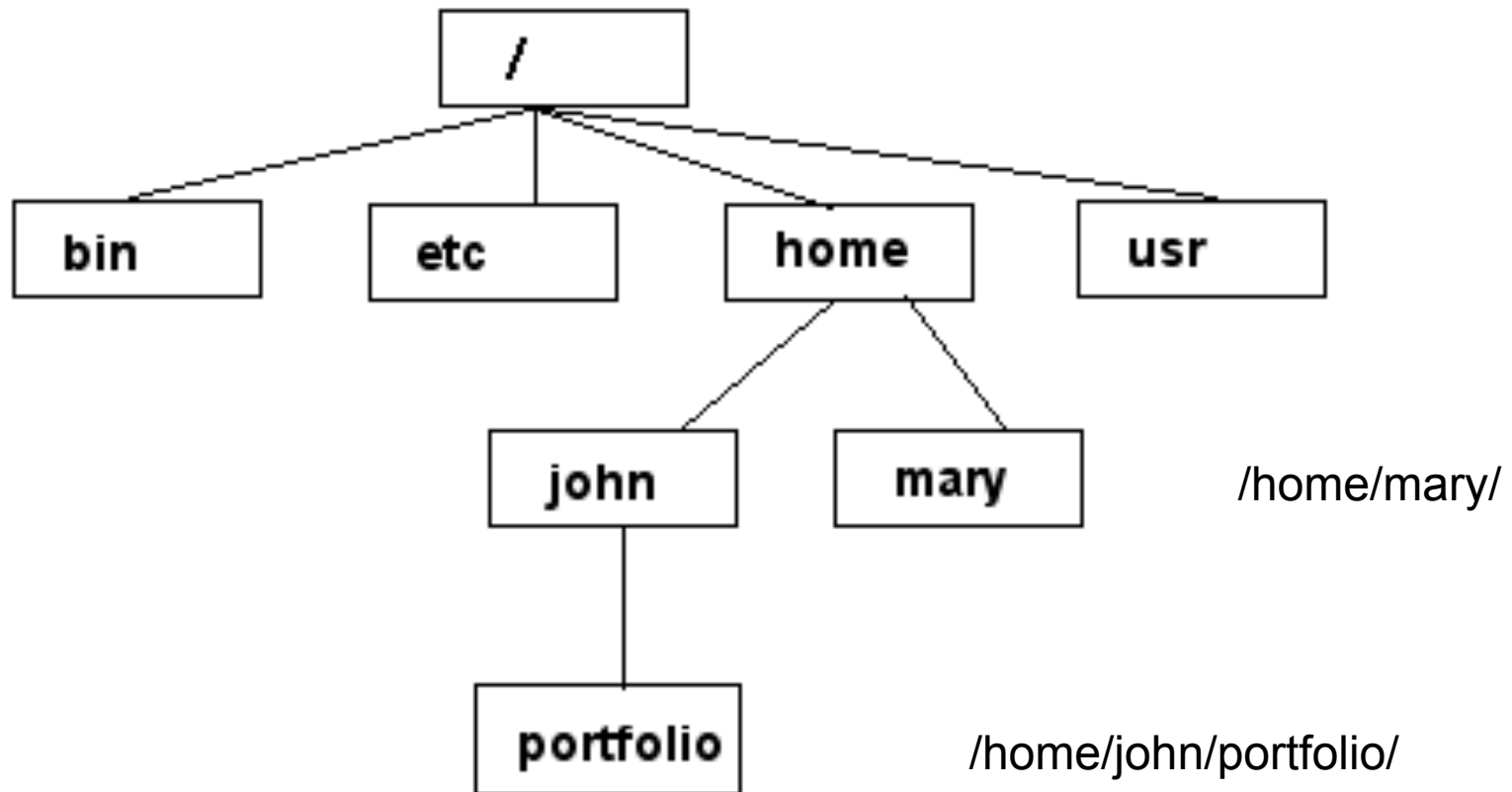
- Unix was the predecessor of Linux
- Linux is a variant of Unix
  - So is Mac OS X, so much of this tutorial applies to Macs as well
- Linux is open source
- Most of the machines you'll use in the **Bioinformatics program** are running the Linux OS

# 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

# Unix/Linux File System

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



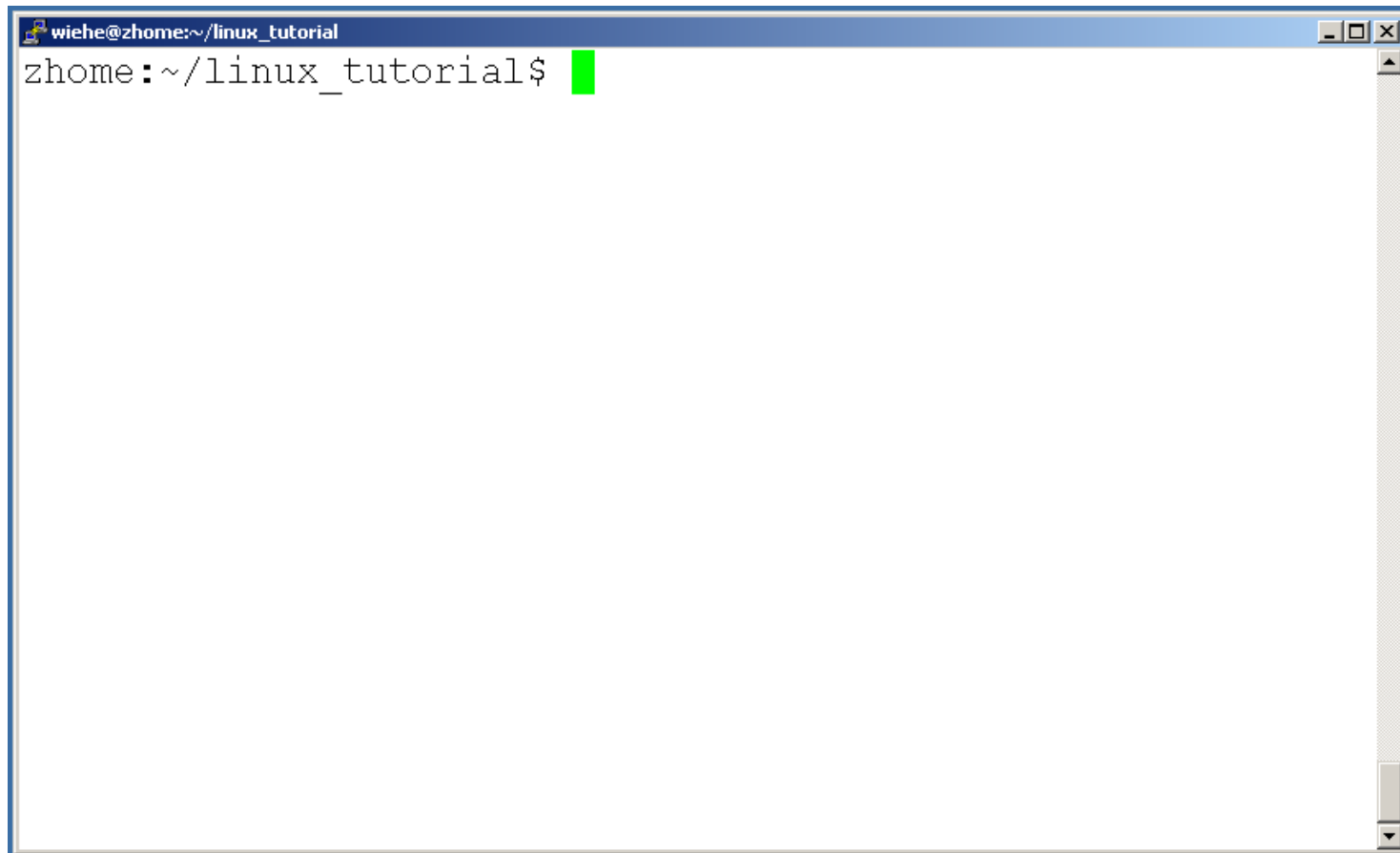
↑  
**The Path**

# 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!**

# Connecting to a Unix/Linux system

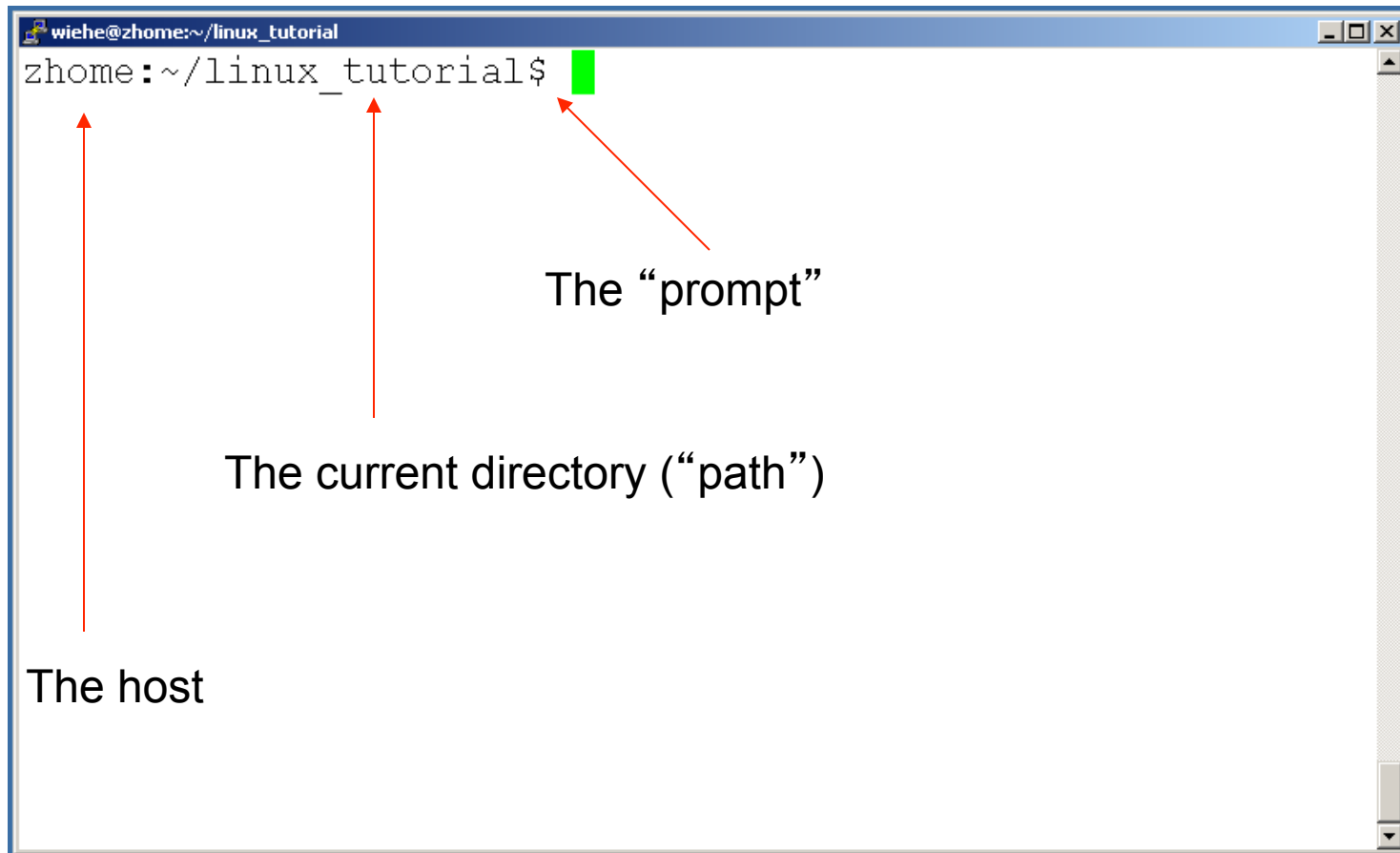
- Open up a terminal:

A screenshot of a terminal window. The title bar at the top reads "wiehe@zhome:~/linux\_tutorial". The main area of the window shows a shell 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 scrollbar on the right side.

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

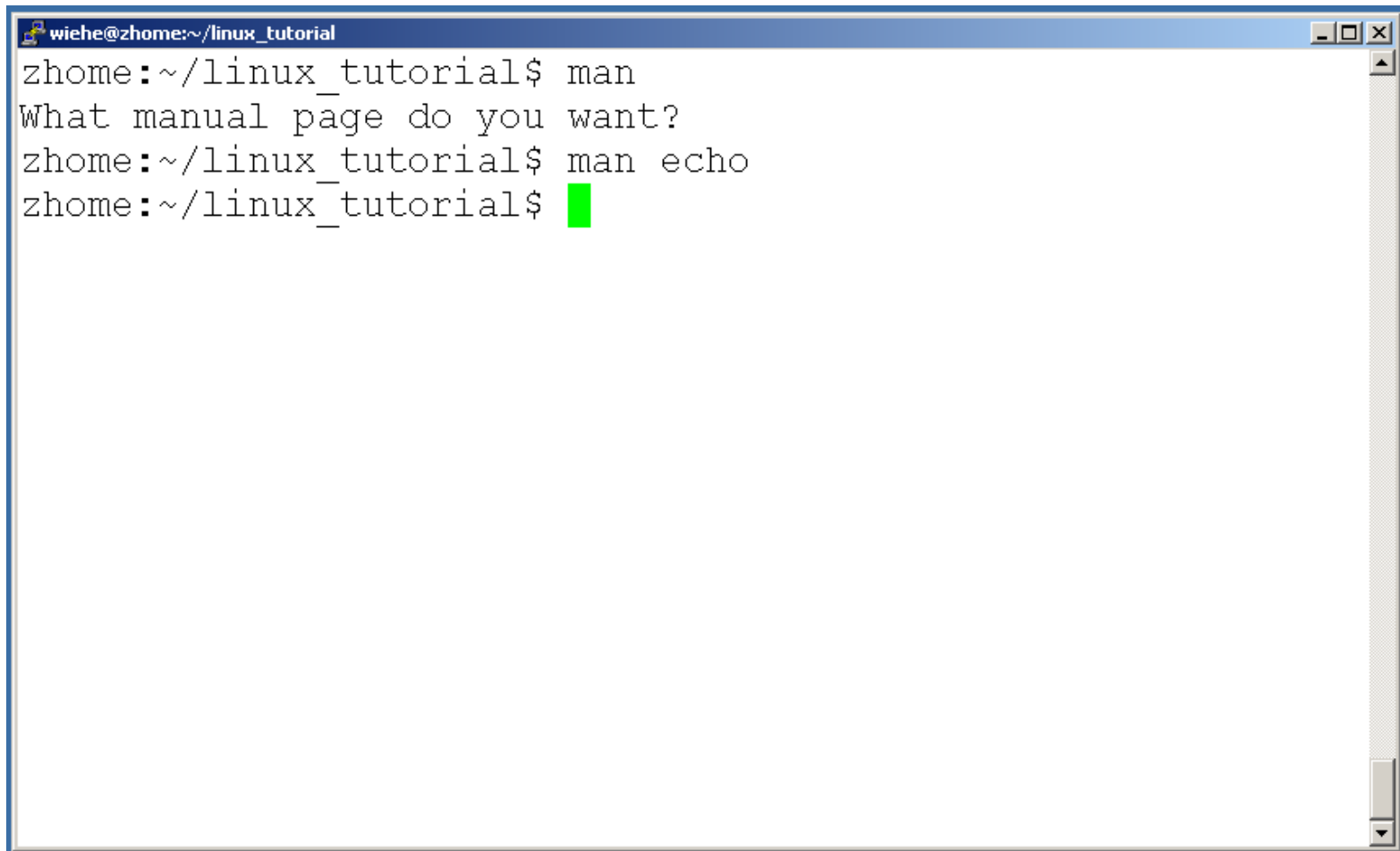
# Connecting to a Unix/Linux system

- Open up a terminal:





# Help!

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux\_tutorial'. The terminal shows a sequence of commands and their outputs. The first command is 'man', which prompts 'What manual page do you want?'. The second command is 'man echo', which is followed by a green cursor block.

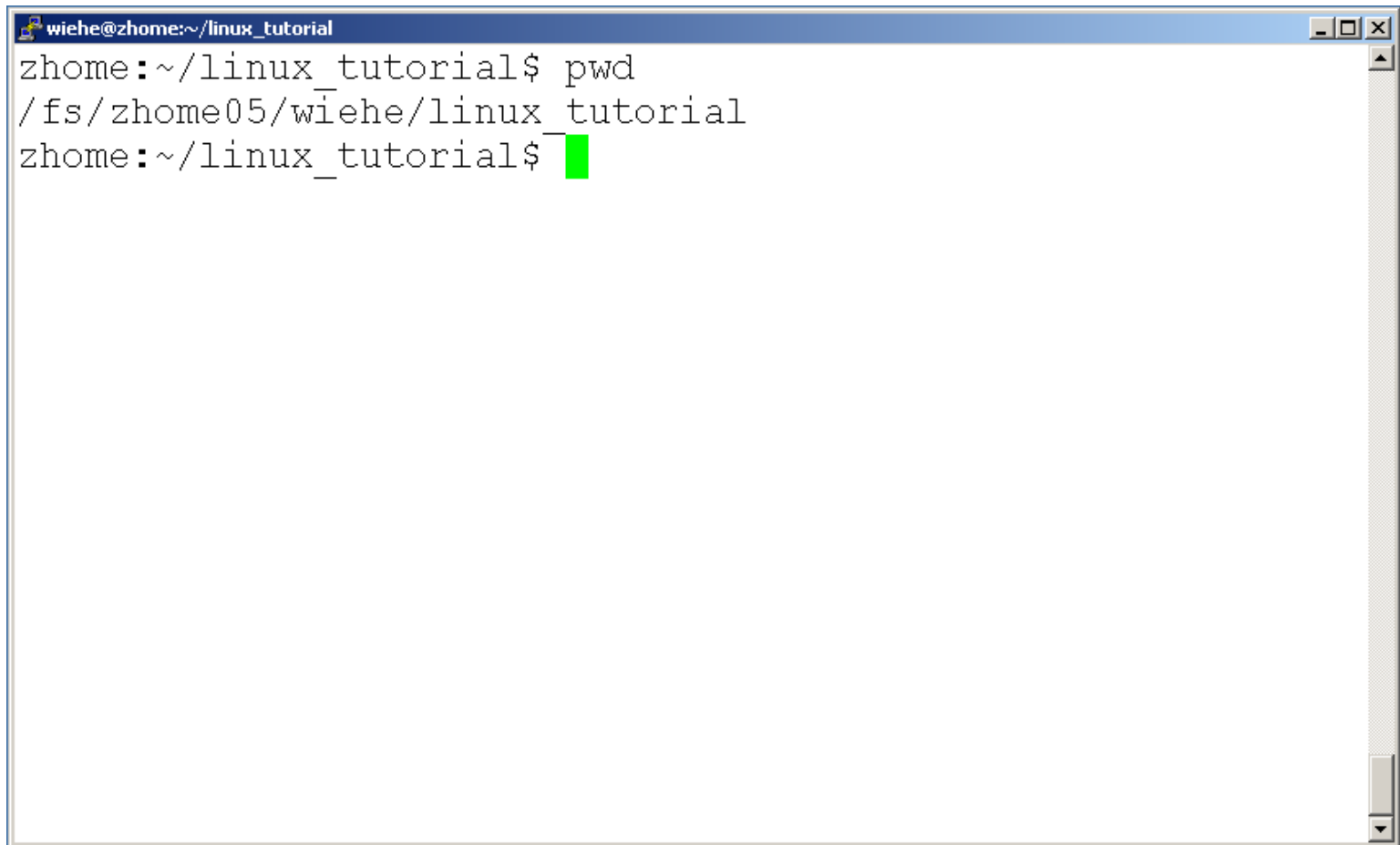
```
wiehe@zhome:~/linux_tutorial$ man
What manual page do you want?
wiehe@zhome:~/linux_tutorial$ man echo
wiehe@zhome:~/linux_tutorial$ █
```

# Help!

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

# Command: pwd

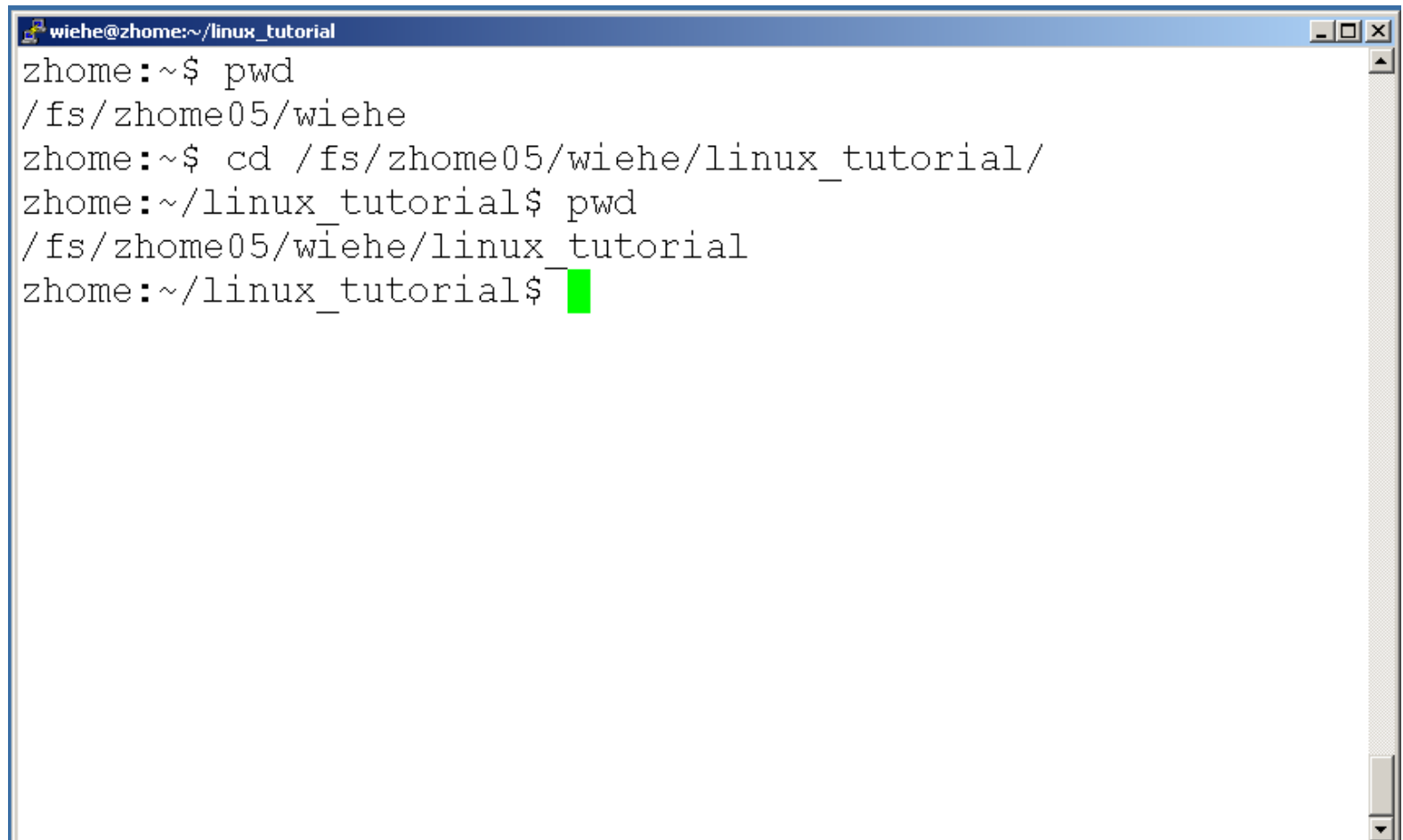
- To find your current path use “pwd”

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux\_tutorial'. The window has standard window controls (minimize, maximize, close) on the right. The terminal content shows a user prompt 'zhome:~/linux\_tutorial\$' followed by the command 'pwd'. The output of the command is '/fs/zhome05/wiehe/linux\_tutorial'. Below the output, the prompt 'zhome:~/linux\_tutorial\$' is shown again with a green cursor block.

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

# Command: cd

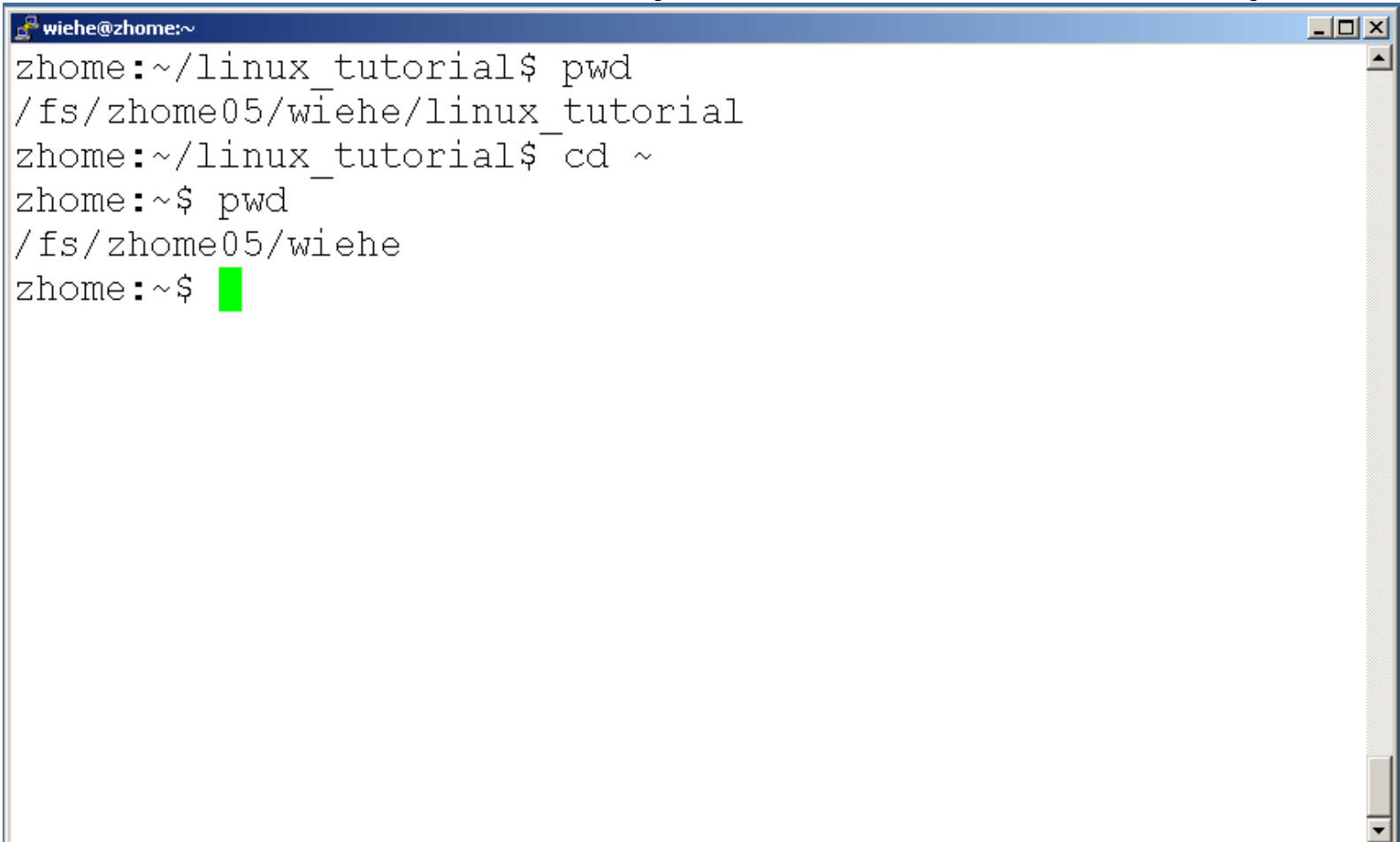
- To change to a specific directory use “cd”

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux\_tutorial'. The window shows a sequence of commands and their outputs. The first command is 'pwd', which outputs '/fs/zhome05/wiehe'. The second command is 'cd /fs/zhome05/wiehe/linux\_tutorial/', which changes the current directory. The third command is 'pwd', which outputs '/fs/zhome05/wiehe/linux\_tutorial'. The prompt is now 'zhome:~/linux\_tutorial\$' followed by a green cursor block.

```
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

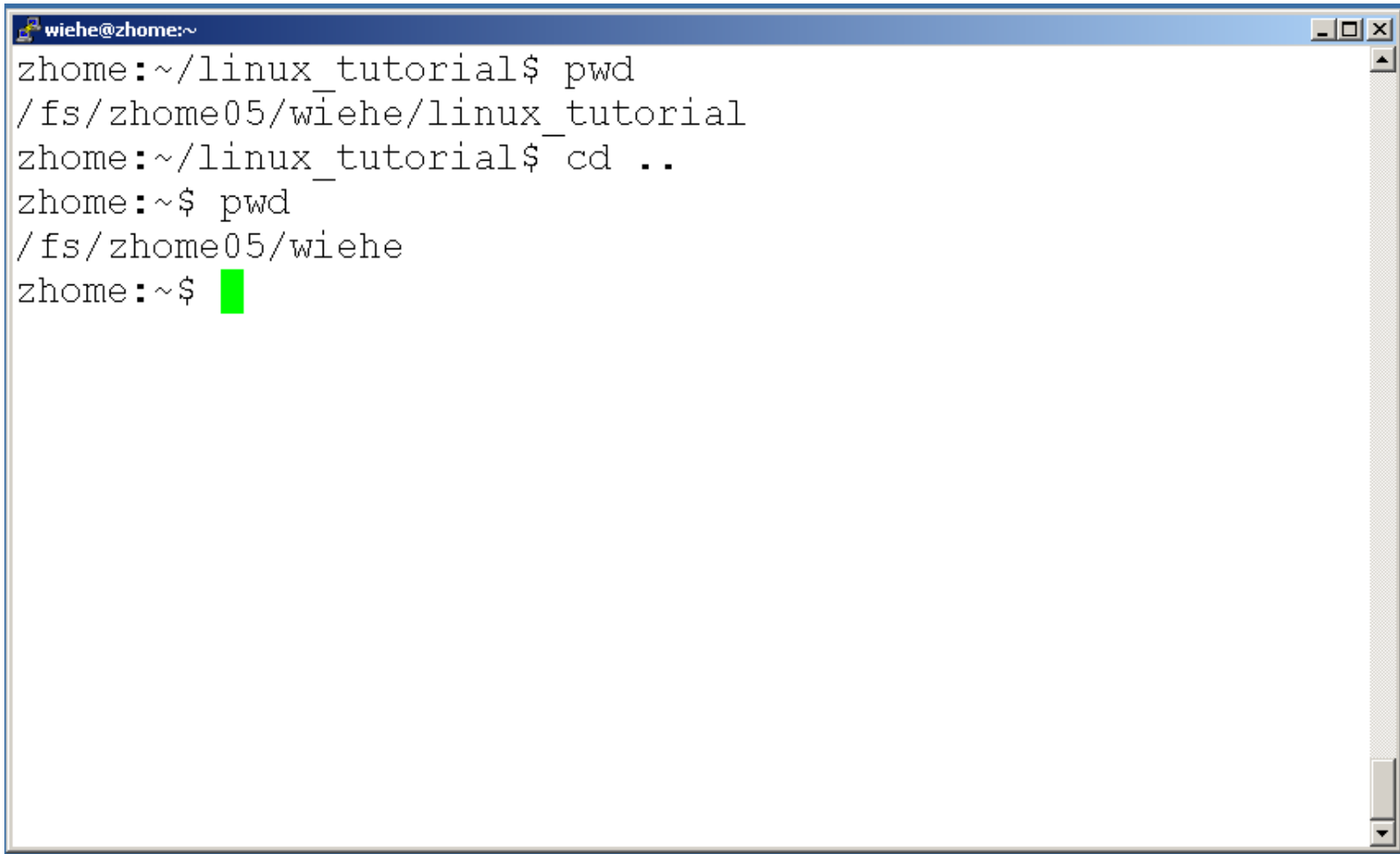
- “~” is the location of your home directory

A terminal window with a blue title bar containing the text 'wiehe@zhome:~'. The terminal shows a sequence of commands and their outputs. The user is initially in the directory '~/linux\_tutorial'. They run 'pwd' and get '/fs/zhome05/wiehe/linux\_tutorial'. Then they run 'cd ~' to move to their home directory. Finally, they run 'pwd' again and get '/fs/zhome05/wiehe'. The prompt 'zhome:~\$' is followed by a green cursor block.

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

# Command: cd

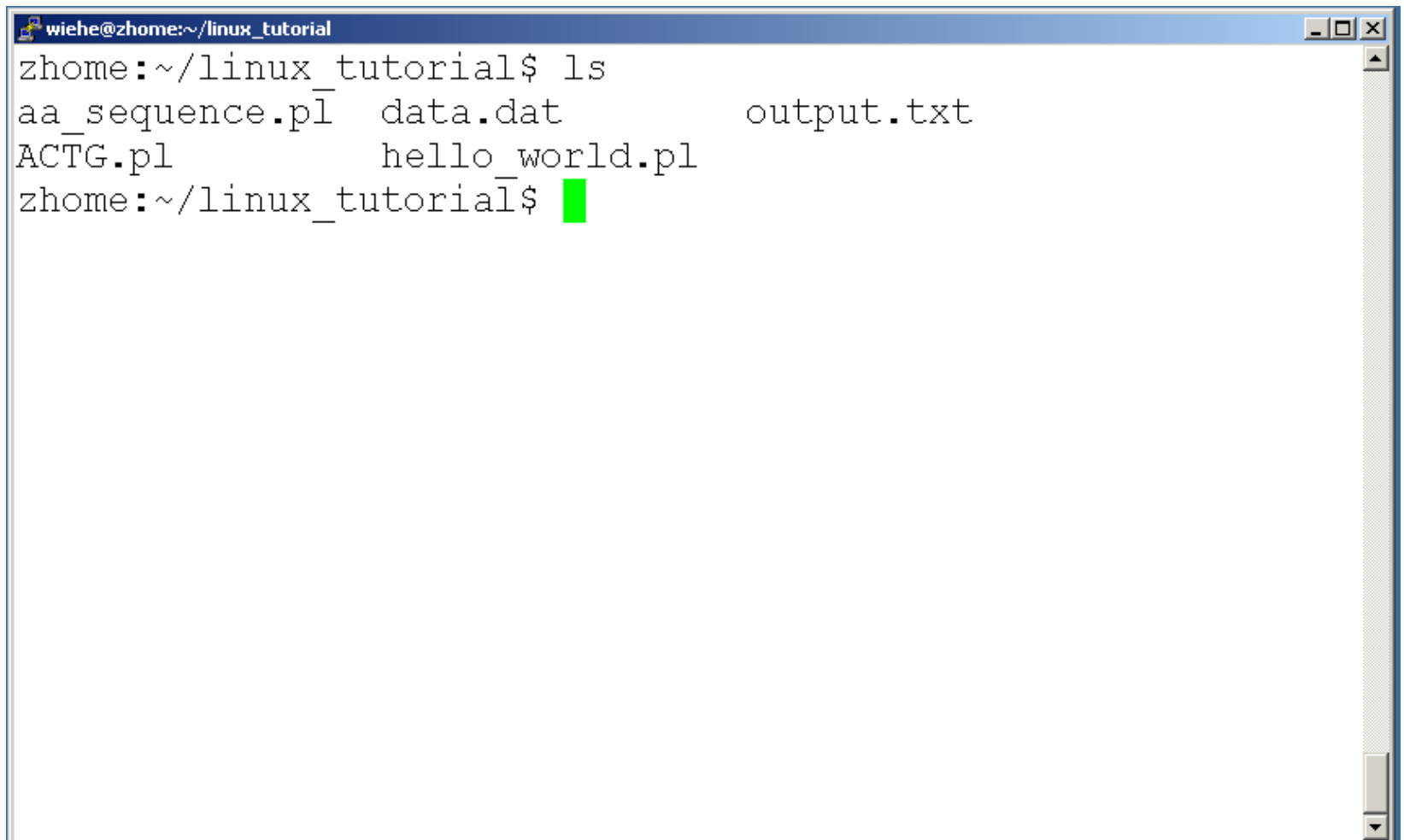
- “..” is the location of the directory below current one

A terminal window titled 'wiehe@zhome:~' with standard window controls. It shows a sequence of commands and their outputs: 'pwd' returns '/fs/zhome05/wiehe/linux\_tutorial', 'cd ..' changes the directory, and a second 'pwd' returns '/fs/zhome05/wiehe'. A green cursor is visible at the end of the final prompt.

```
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 screenshot of a Linux terminal window. The title bar at the top reads 'wiehe@zhome:~/linux\_tutorial'. The terminal shows the command 'ls' being executed, which lists the files 'aa\_sequence.pl', 'data.dat', 'output.txt', and 'hello\_world.pl'. The prompt 'zhome:~/linux\_tutorial\$' is shown at the bottom, 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$
```

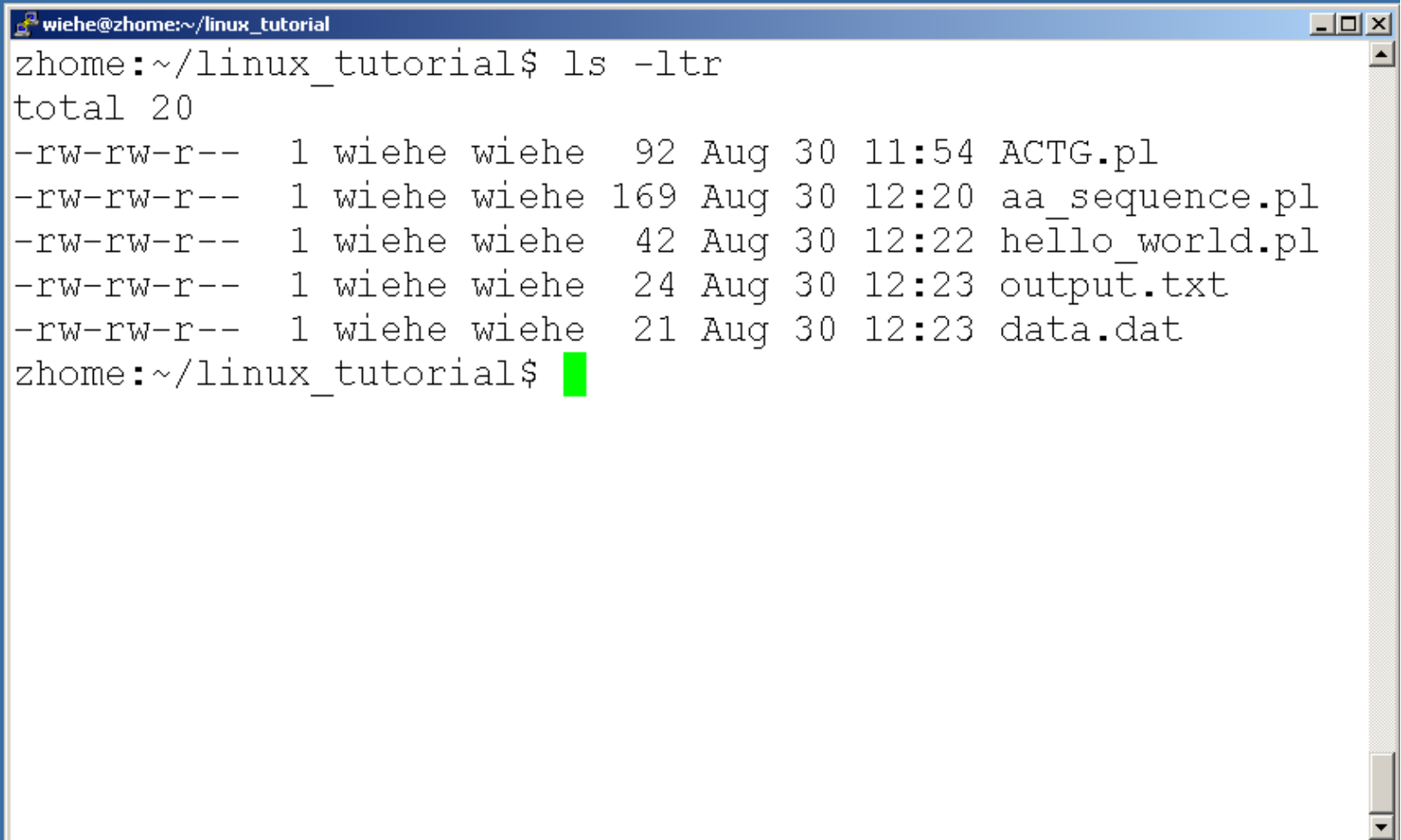
# Command: ls

- 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

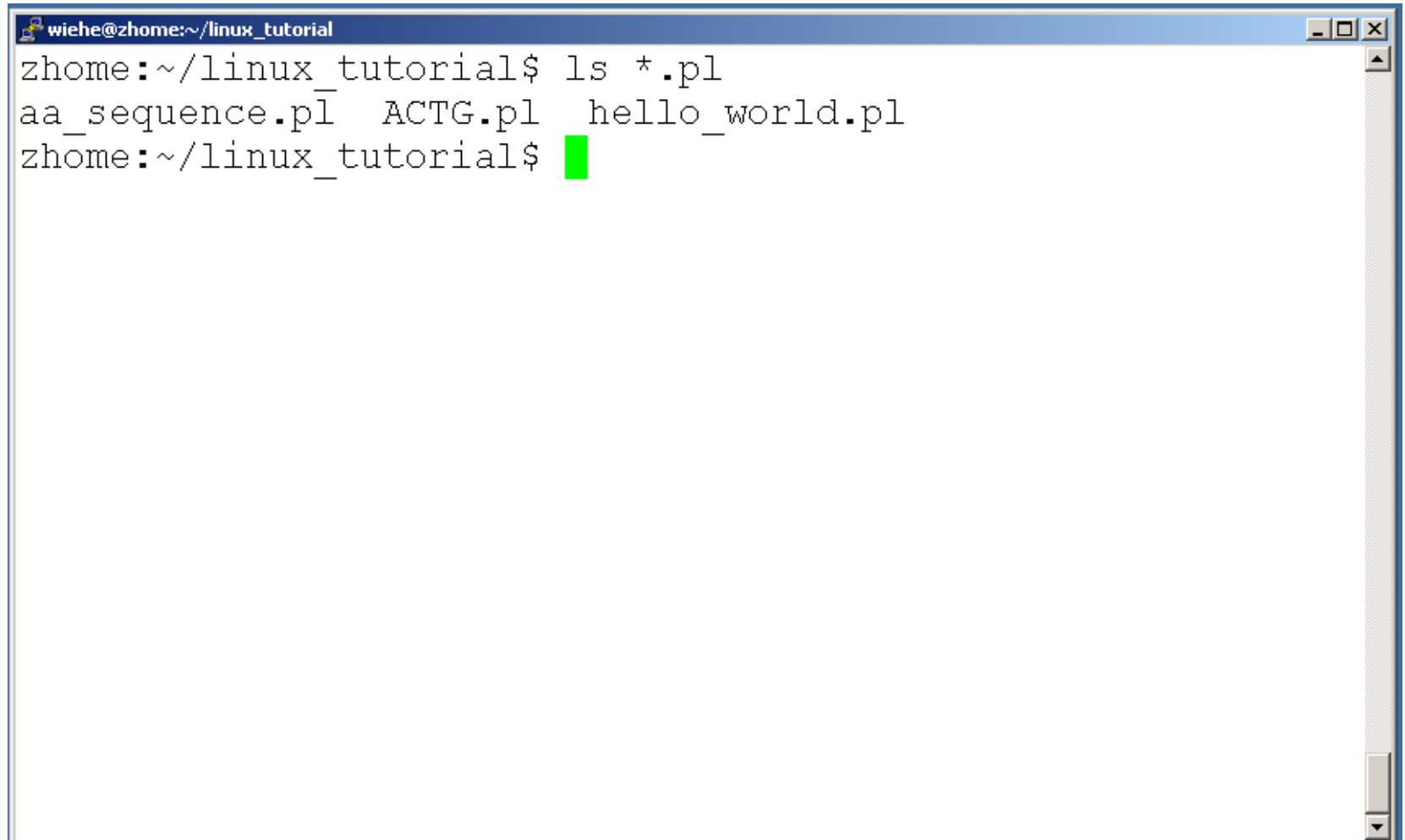
- List files by time in reverse order with long listing

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux\_tutorial'. The window shows the command 'ls -ltr' being executed. 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). Each line starts with permissions '-rw-rw-r--', followed by the number of links '1', the owner 'wiehe', the group 'wiehe', the file size, the date and time, and the filename. 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: \*

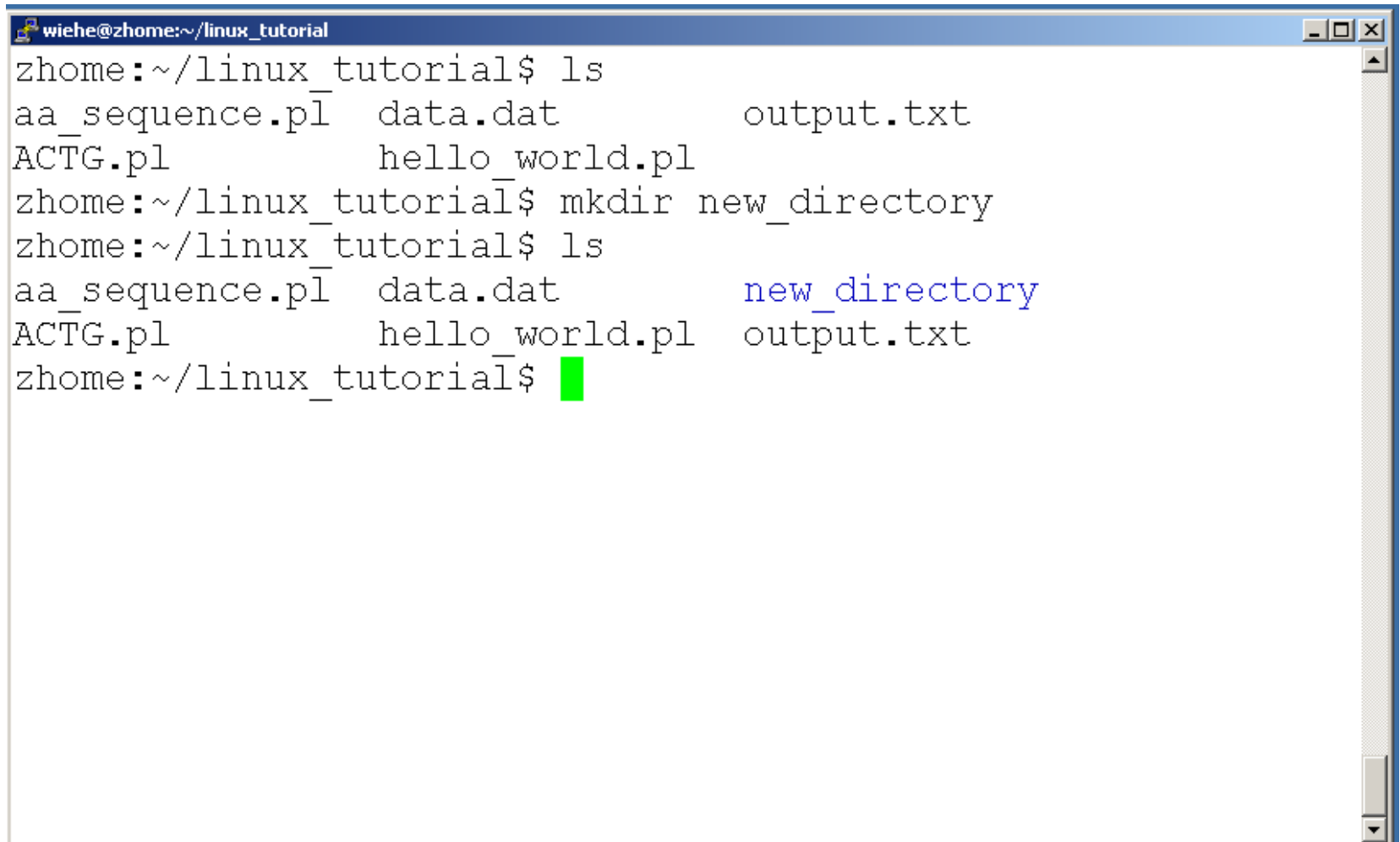
- “\*” can be used as a wildcard in unix/linux

A screenshot of a terminal window with a blue title bar. The title bar text is 'wiehe@zhome:~/linux\_tutorial'. The terminal content shows a command 'ls \*.pl' being executed, which lists three files: 'aa\_sequence.pl', 'ACTG.pl', and 'hello\_world.pl'. Below the list, the prompt 'zhome:~/linux\_tutorial\$' is followed by 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

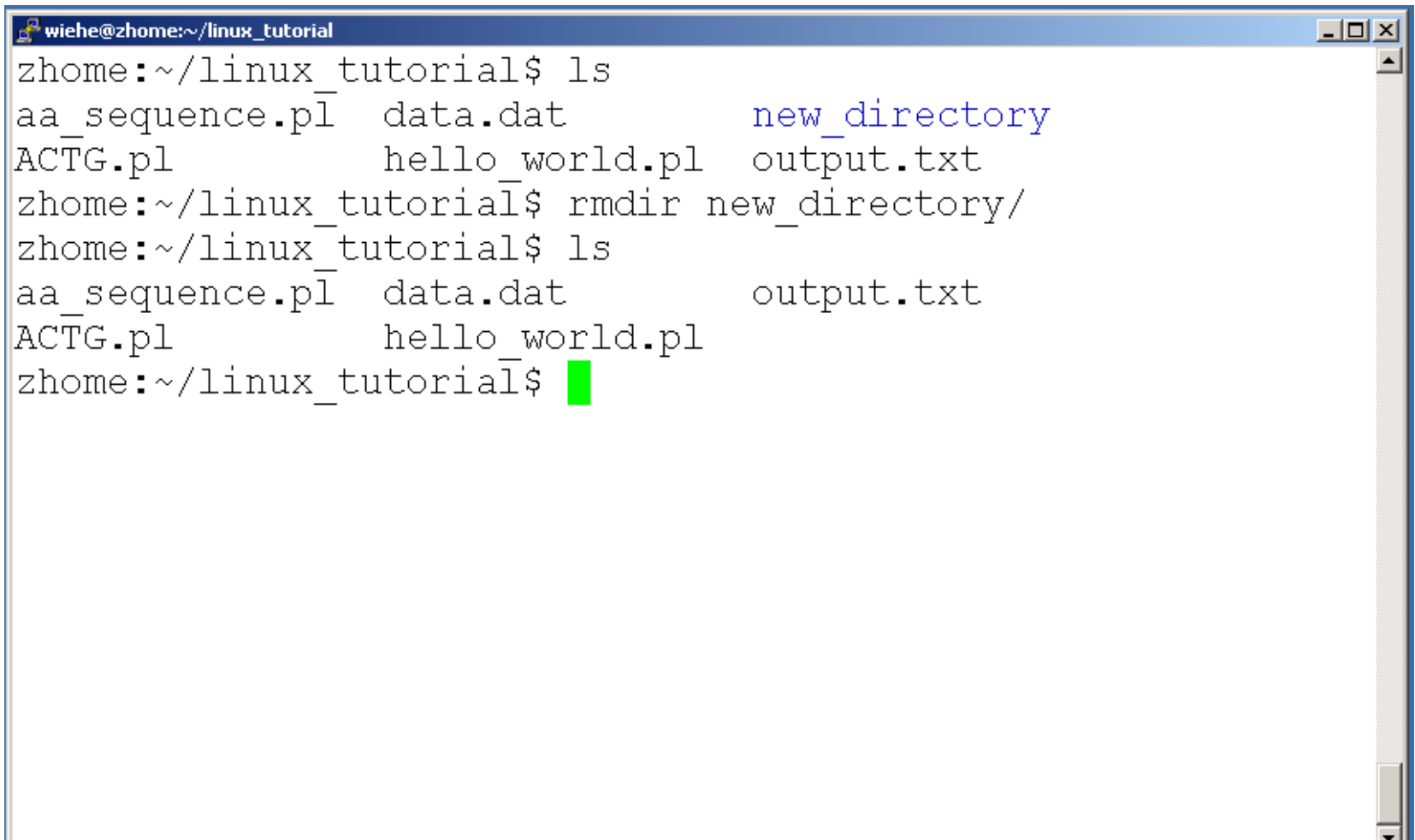
- To create a new directory use “mkdir”

A terminal window titled 'wiehe@zhome:~/linux\_tutorial' with standard window controls. It shows a sequence of commands and their outputs. First, 'ls' is run, listing files: 'aa\_sequence.pl', 'data.dat', 'output.txt', and 'ACTG.pl'. Then, 'mkdir new\_directory' is executed. Finally, 'ls' is run again, showing the same files plus 'new\_directory' in blue text. The prompt ends with 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$ 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$
```

# Creating files in Unix/Linux

- Requires the use of an Editor
- Various Editors:
  - 1) Nano
  - 2) vi
  - 3) emacs



# 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

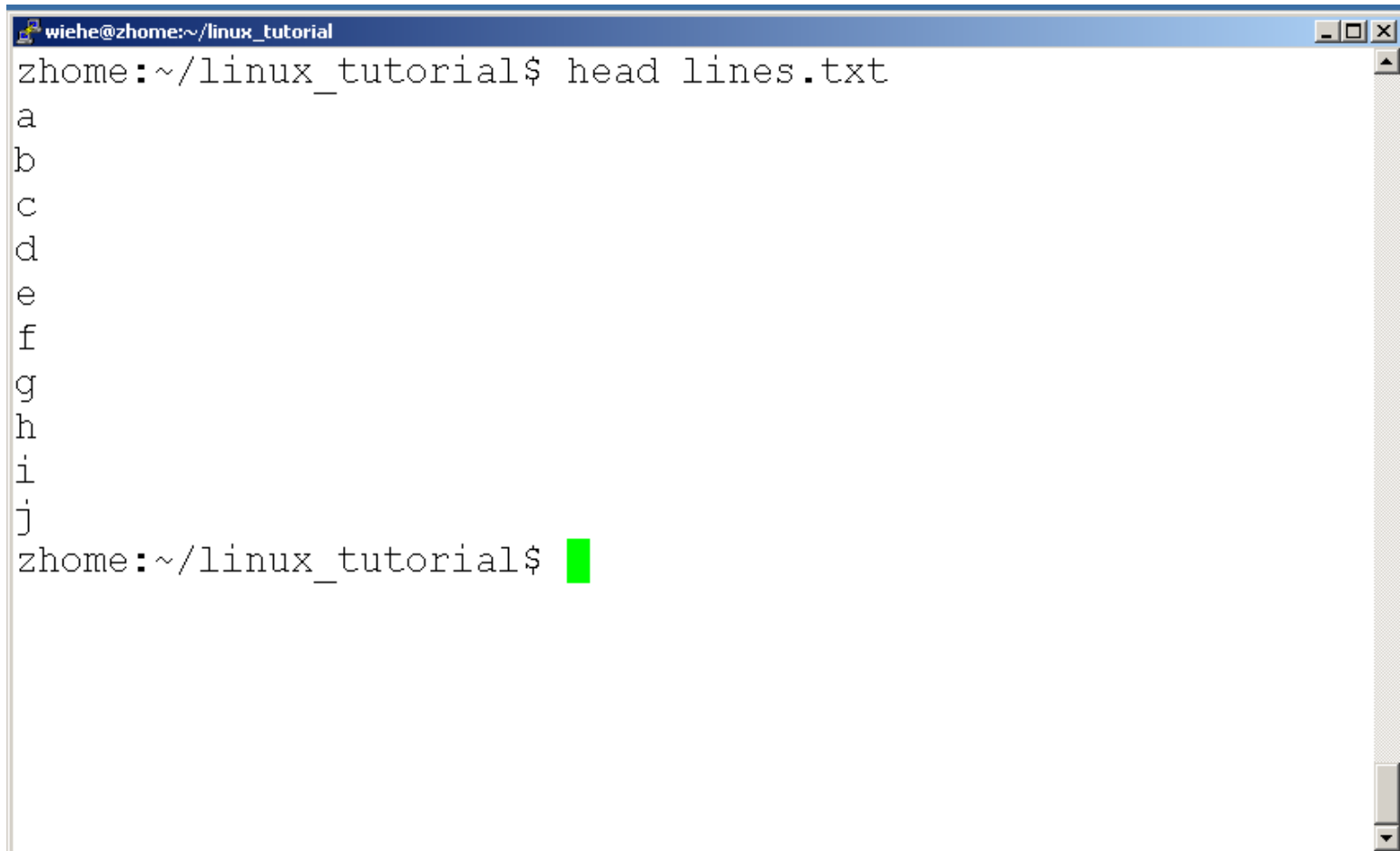


# Command: head

- “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

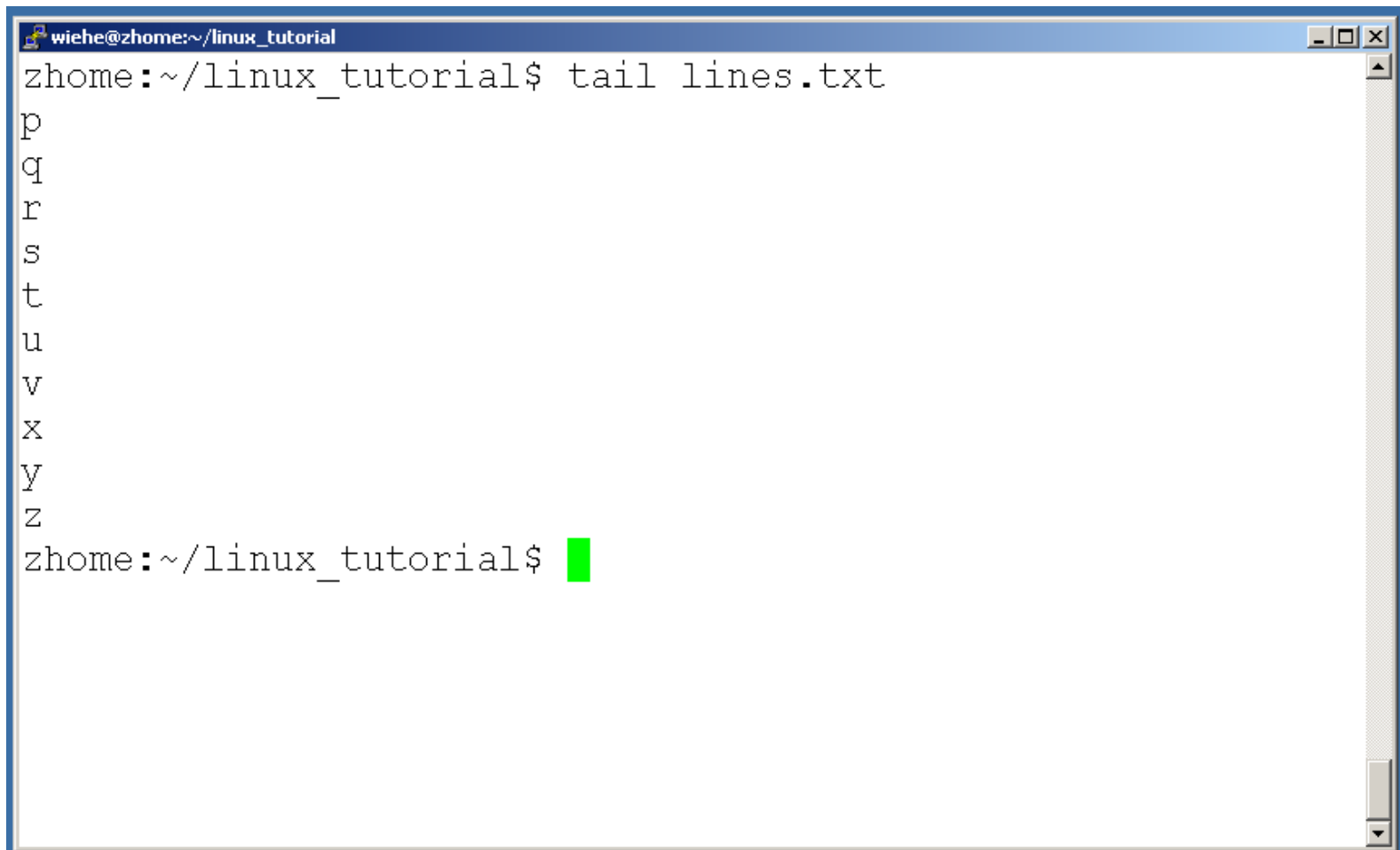
- Here's an example of using “head”:

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, which outputs the first ten lines of a file: 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', and 'j'. The prompt 'zhome:~/linux\_tutorial\$' is shown again with a green cursor block.

```
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. Below the last line of output, the prompt 'zhome:~/linux\_tutorial\$' is followed by a green cursor block.

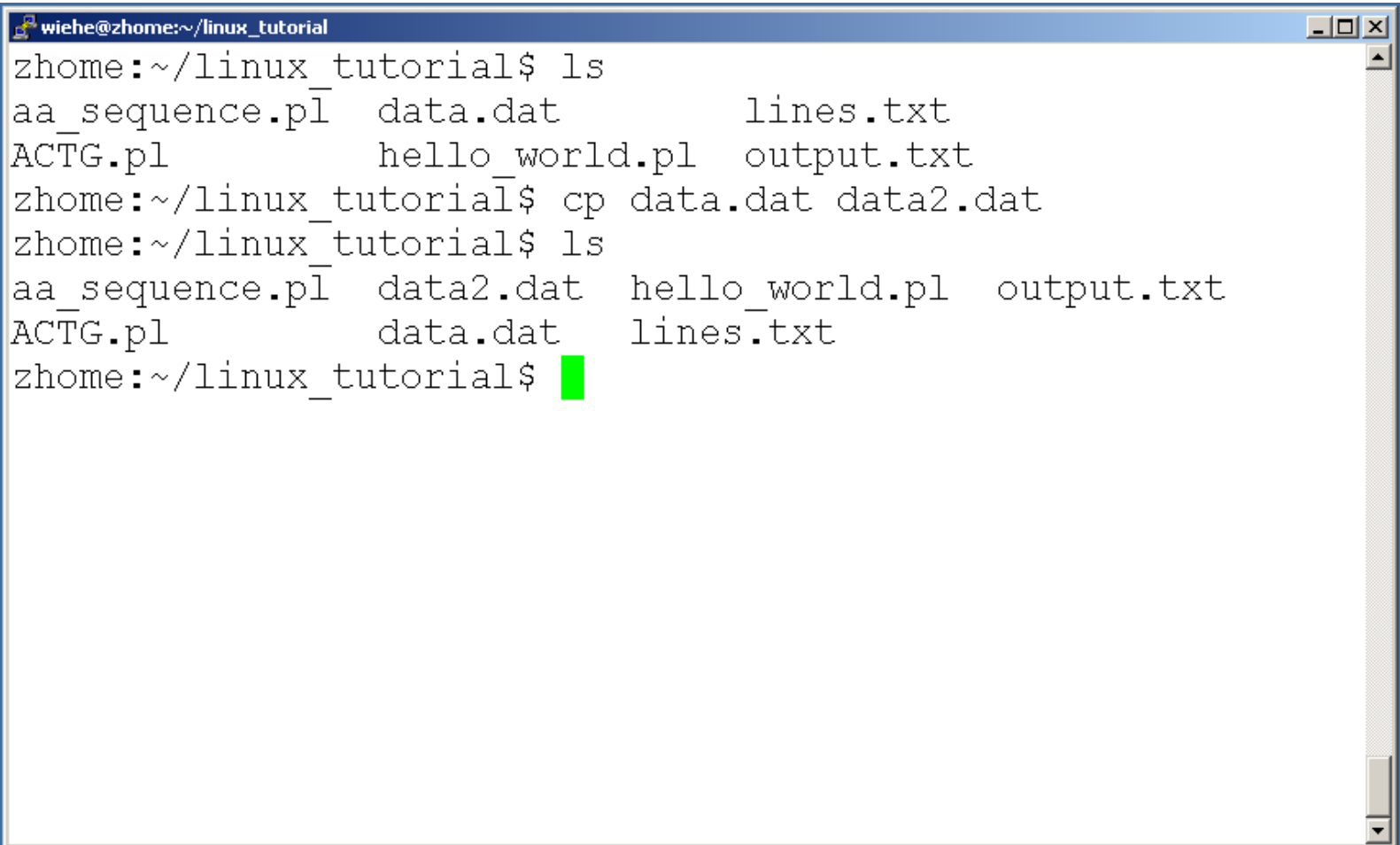
```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ tail lines.txt
p
q
r
s
t
u
v
x
y
z
zhome:~/linux_tutorial$
```

# File Commands

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

# Command: cp

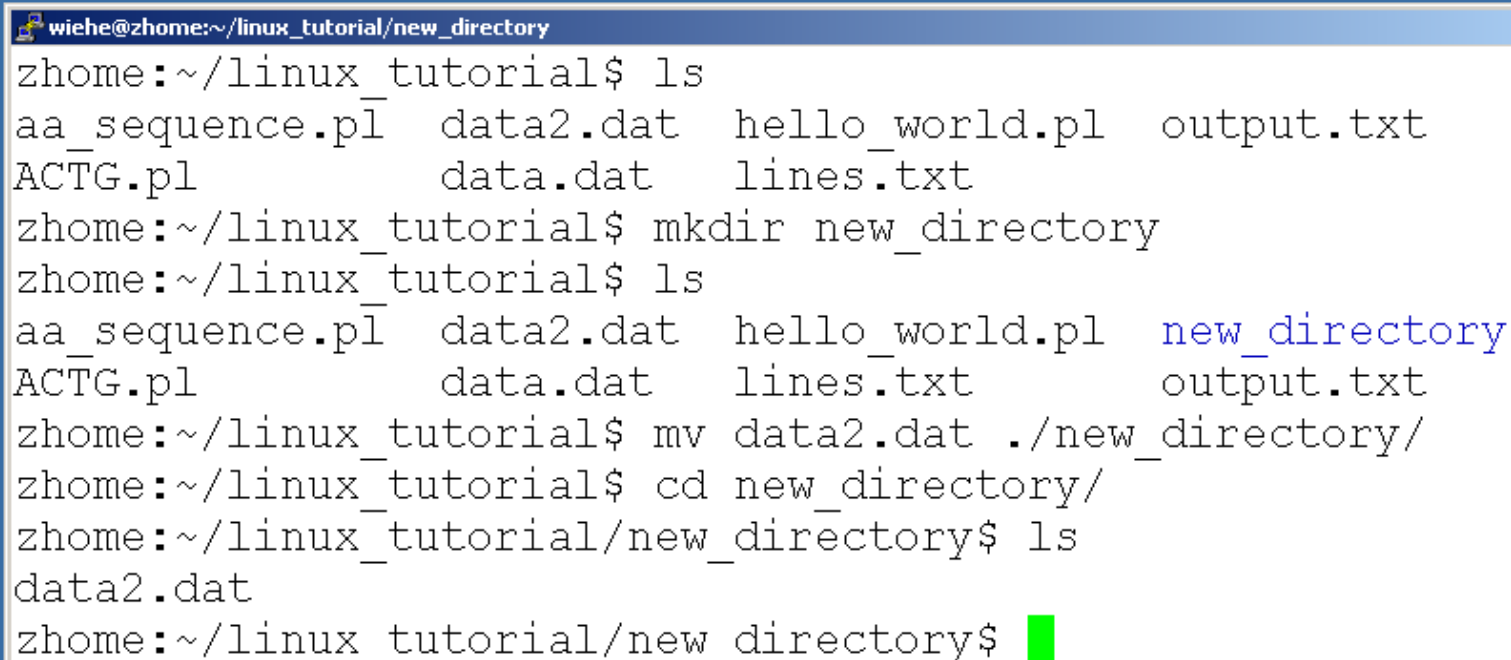
- To copy a file use “cp”

A terminal window titled 'wiehe@zhome:~/linux\_tutorial' with standard window controls. The terminal shows a sequence of commands and their outputs. First, 'ls' is run, listing 'aa\_sequence.pl', 'data.dat', and 'lines.txt'. Then, 'cp data.dat data2.dat' is executed. Finally, 'ls' is run again, showing 'aa\_sequence.pl', 'data2.dat', 'hello\_world.pl', and 'output.txt', while 'data.dat' and 'lines.txt' are no longer listed. A green cursor is at the end of the last prompt.

```
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

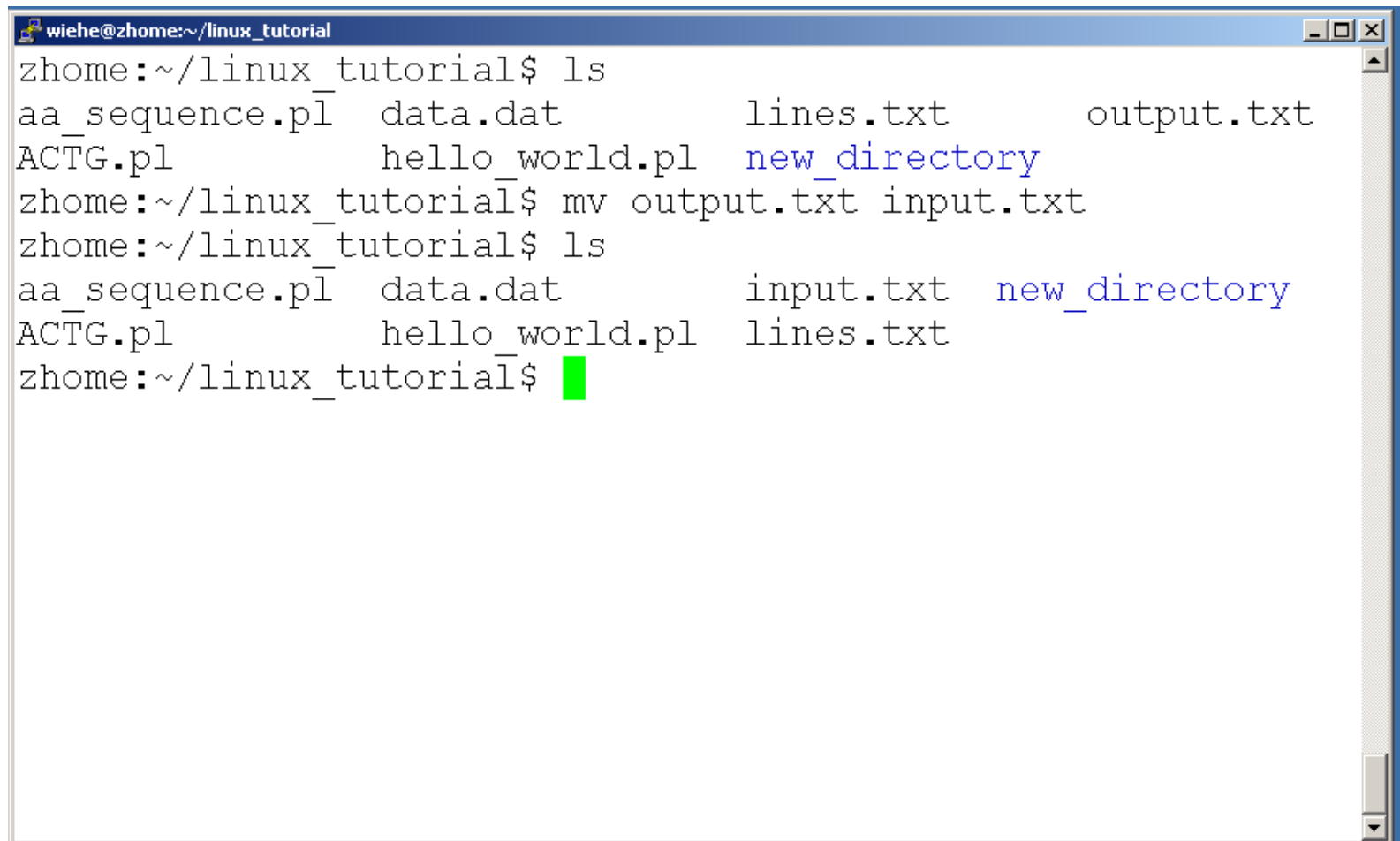
- To move a file to a different location use “mv”

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux\_tutorial/new\_directory'. The terminal shows a series of commands and their outputs. The 'ls' command is used twice to show the directory contents before and after moving a file. The 'mkdir' command is used to create a new directory. The 'mv' command is used to move 'data2.dat' to the 'new\_directory' subdirectory. The 'cd' command is used to navigate into the 'new\_directory'. The final 'ls' command shows 'data2.dat' as the only file in the current directory.

```
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

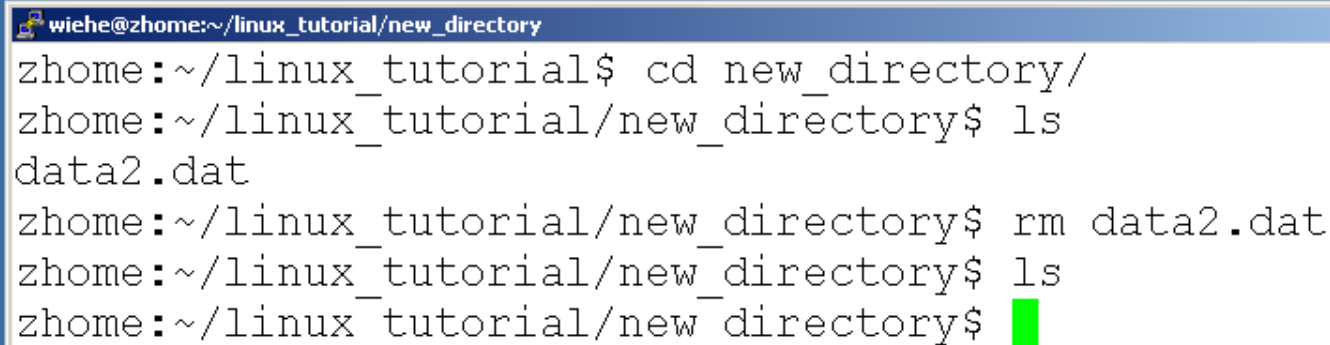
- mv can also be used to rename a file



```
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$
```

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/" changes the directory; "ls" lists the contents, showing "data2.dat"; "rm data2.dat" removes the file; and a second "ls" command is executed, resulting in a blank line. A green cursor is visible at the end of the final prompt.



# 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**

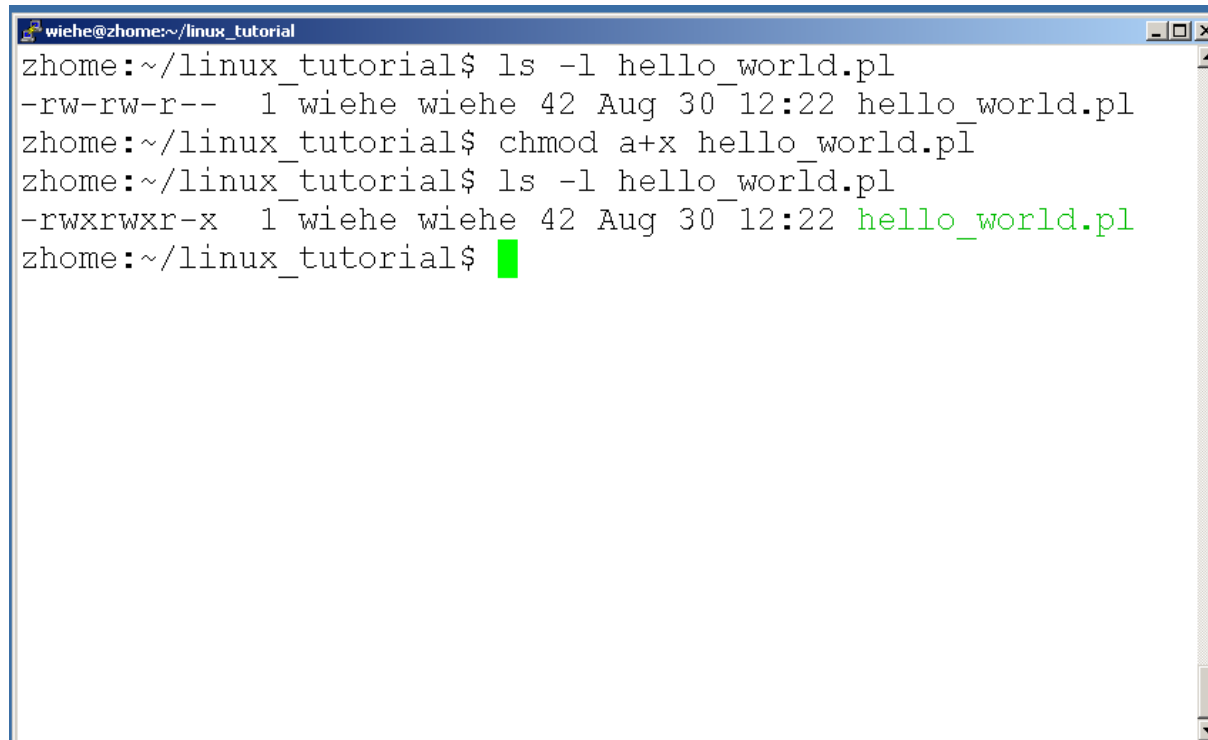
# File Permissions

```
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 [user/group/others/all]+[permission] [file(s)]`
  - Below we grant execute permission to all:

A terminal window titled 'wiehe@zhome:~/linux\_tutorial' showing a sequence of commands and their outputs. The first command is 'ls -l hello\_world.pl', which outputs '-rw-rw-r-- 1 wiehe wiehe 42 Aug 30 12:22 hello\_world.pl'. The second command is 'chmod a+x hello\_world.pl'. The third command is 'ls -l hello\_world.pl', which outputs '-rwxrwxr-x 1 wiehe wiehe 42 Aug 30 12:22 hello\_world.pl', with 'hello\_world.pl' highlighted in green. The prompt 'zhome:~/linux\_tutorial\$' is followed by a green cursor.

```
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



# 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

# 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”