

Plan for today

- ▶ From raw reads to “R ready” bam files
 - ▶ Part I: Introduction to Linux
 - ▶ Part II: Working with NGS data



Introduction to Linux

Adapted from Eric Bishop

10th August 2012

What is Linux?

- ▶ Just another operating system
- ▶ Linux is open source
- ▶ Linux is a variant of Unix
 - ▶ So is Mac OS X, so much of this tutorial applies to Macs as well



Introduction: Why Linux?

- ▶ Linux is **free**
- ▶ It's fully **customizable**
- ▶ It's **stable** (i.e. it almost never crashes)
- ▶ Bla, bla, bla

- ▶ But: People are used to Windows/Mac!

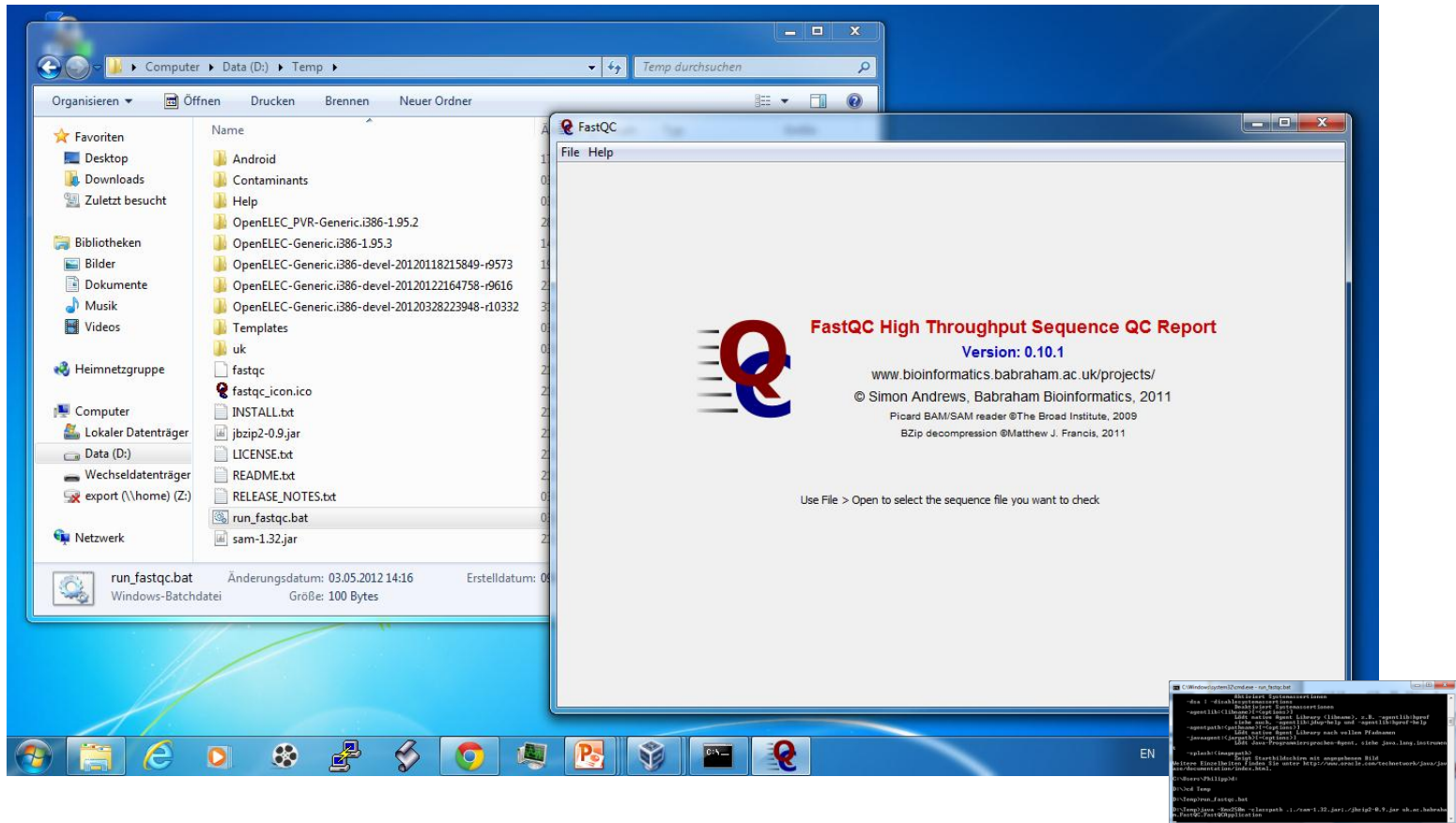


So why Linux?

- ▶ When working with NGS data you don't really have another choice
 - ▶ It is possible to handle extremely **large files** without problems
 - ▶ Most of the **software** was developed/optimized for Linux
 - ▶ It is easy to run programs on **remote machines**



GUI vs Shell: Windows



- ▶ Windows: focus on graphical user interface

GUI vs Shell: Linux

```
philipp@newton:/scratch/philipp> l
total 13G
drwxr-xr-x 19 philipp_CIBIV 4.0K Aug 9 08:55 .
drwxrwxrwt 4 root root 34 Dec 21 2011 ..
drwxr-x--- 7 philipp_CIBIV 101 May 9 09:19 android-sdk-linux
-rw-r--r-- 1 philipp_CIBIV 29M Apr 9 20:00 android-sdk_r18-linux.tgz
drwxr-xr-x 4 philipp_CIBIV 31 Mar 5 10:29 backup
drwxr-xr-x 2 philipp_CIBIV 24 Aug 3 10:33 bin
drwxr-xr-x 7 philipp_CIBIV 77 Dec 21 2011 bin_backup
-rwxr-xr-x 1 philipp_CIBIV 3.9K Feb 28 17:06 callSNPs_test.sh
-rw-r--r-- 1 philipp_CIBIV 44K Aug 3 10:45 compmotives.tar.gz
drwxr-xr-x 9 philipp_CIBIV 4.0K Apr 2 08:37 eclipse
drwxr-xr-x 9 philipp_CIBIV 4.0K Apr 2 08:37 eclipse-cpp
drwxr-xr-x 9 philipp_CIBIV 4.0K Aug 8 08:22 eclipse-x64
drwxr-xr-x 7 philipp_CIBIV 4.0K Feb 27 20:27 e-coli-k12-resequencing-1.3.0
-rw-r--r-- 1 philipp_CIBIV 768M Mar 2 13:26 e-coli-k12-resequencing-1.3.0.tgz.cdndownload
-rw-r--r-- 1 philipp_CIBIV 4.8G Aug 3 12:31 ERR039480.fastq.gz
drwxr-xr-x 7 philipp_CIBIV 64 Mar 2 10:42 gdb
drwxr-xr-x 15 philipp_CIBIV 4.0K Mar 2 10:30 gdb-7.4
-rw-r--r-- 1 philipp_CIBIV 27M Jan 24 2012 gdb-7.4.tar.gz
drwxr-xr-x 2 philipp_CIBIV 66 Apr 27 09:56 lib64
drwxr-xr-x 5 philipp_CIBIV 74 Dec 21 2011 ngm-0.2.0
drwxr-xr-x 6 philipp_CIBIV 4.0K May 21 08:45 NGM_backup_20120521
drwxr-xr-x 3 philipp_CIBIV 17 Jul 10 08:54 ngm-core-backup
-rw-r--r-- 1 philipp_CIBIV 161K Dec 21 2011 ngm-src-0.2.0.tar.bz2
-rwxr-xr-x 1 philipp_CIBIV 74K Aug 10 2011 NGM_TestSuite
-rw-r--r-- 1 philipp_CIBIV 1.6G Aug 9 08:25 reads.fa
drwxr-xr-x 3 philipp_CIBIV 118 Aug 8 11:42 r-kurs
drwxr-xr-x 3 philipp_CIBIV 43 Oct 20 2011 simulated_data
-rw-r--r-- 1 philipp_CIBIV 1.3K Apr 2 09:19 test.fa
-rw-r--r-- 1 philipp_CIBIV 2.3G Aug 9 09:03 test.sam
drwxr-xr-x 5 philipp_CIBIV 4.0K Jun 28 11:34 tmp
philipp@newton:/scratch/philipp>
```

- ▶ Linux: focus on command line interface (GUI just an „Addon“)

What exactly is a “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!**

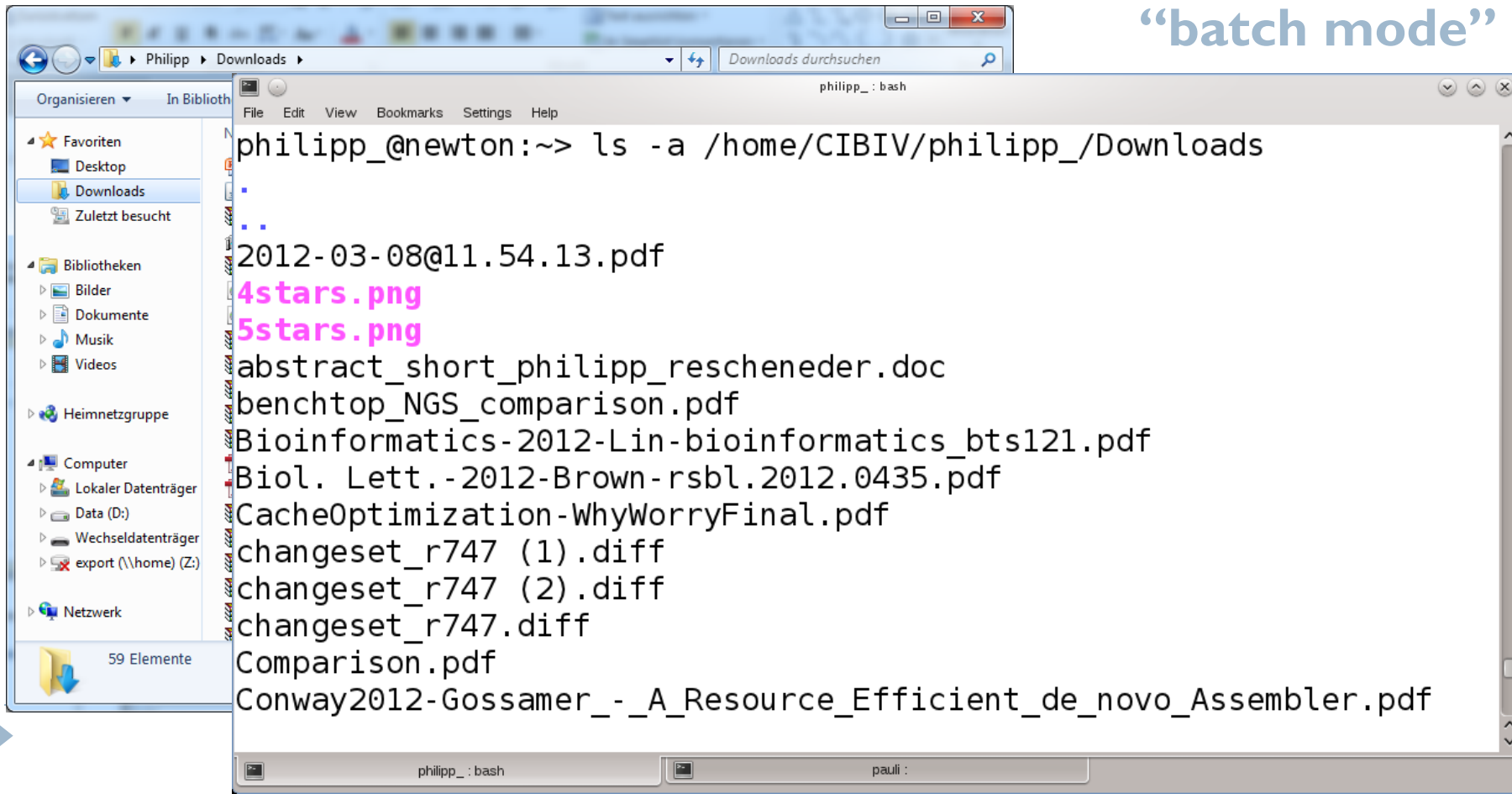


GUI vs Shell: Example

- ▶ Show (hidden) files in Download folder

“Interactive”

“batch mode”



The screenshot displays a Linux desktop environment. On the left, a file manager window shows the 'Downloads' folder. The file list includes:

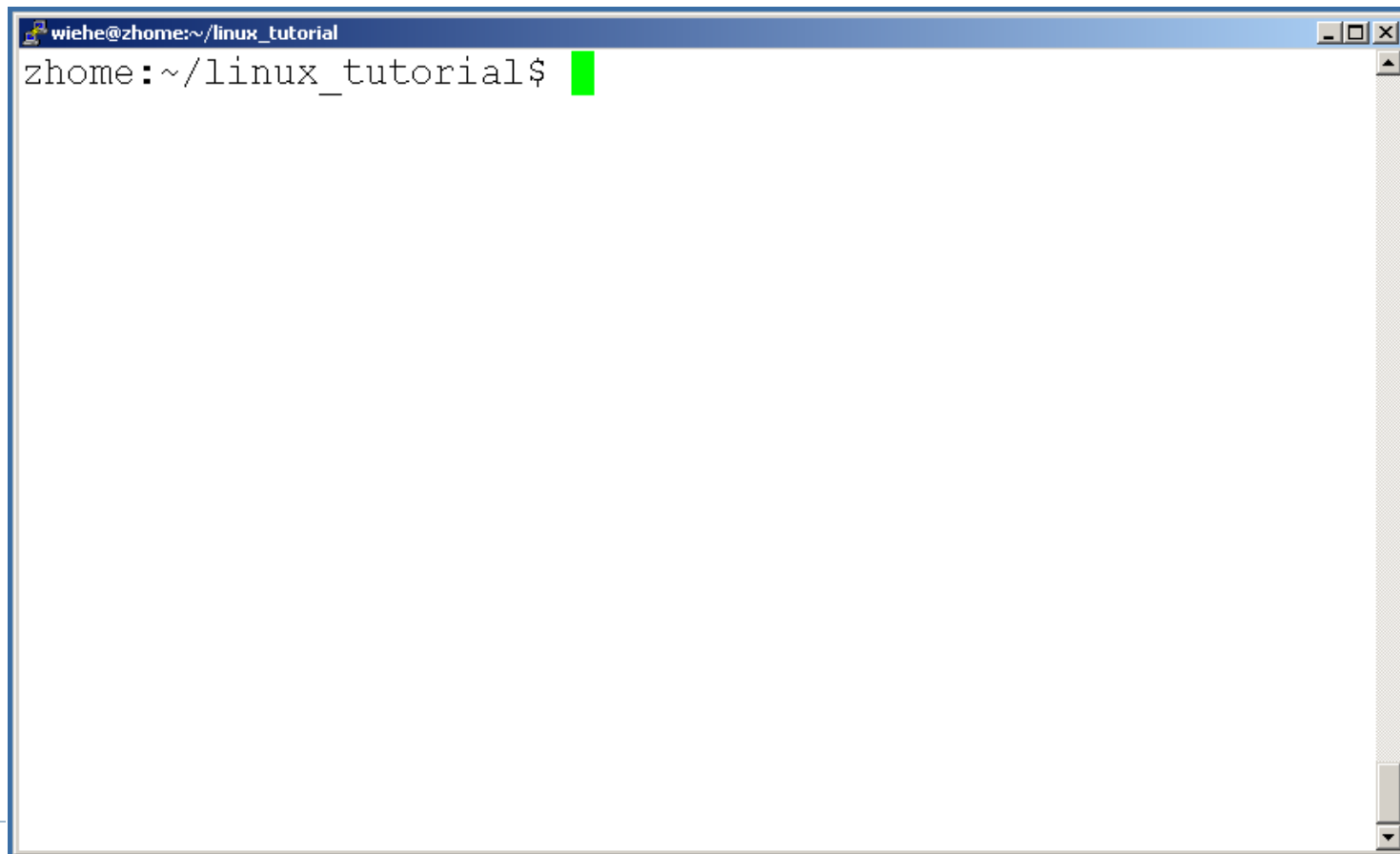
- 2012-03-08@11.54.13.pdf
- .4stars.png
- .5stars.png
- abstract_short_philipp_rescheneder.doc
- benchtop_NGS_comparison.pdf
- Bioinformatics-2012-Lin-bioinformatics_bts121.pdf
- Biol. Lett.-2012-Brown-rsbl.2012.0435.pdf
- CacheOptimization-WhyWorryFinal.pdf
- changeset_r747 (1).diff
- changeset_r747 (2).diff
- changeset_r747.diff
- Comparison.pdf
- Conway2012-Gossamer_-_A_Resource_Efficient_de_novo_Assembler.pdf

On the right, a terminal window titled 'philipp_: bash' shows the command `ls -a /home/CIBIV/philipp_/Downloads` and its output, which matches the file manager's list. The terminal output is:

```
philipp_@newton:~> ls -a /home/CIBIV/philipp_/Downloads
.
..
2012-03-08@11.54.13.pdf
.4stars.png
.5stars.png
abstract_short_philipp_rescheneder.doc
benchtop_NGS_comparison.pdf
Bioinformatics-2012-Lin-bioinformatics_bts121.pdf
Biol. Lett.-2012-Brown-rsbl.2012.0435.pdf
CacheOptimization-WhyWorryFinal.pdf
changeset_r747 (1).diff
changeset_r747 (2).diff
changeset_r747.diff
Comparison.pdf
Conway2012-Gossamer_-_A_Resource_Efficient_de_novo_Assembler.pdf
```

Connecting to a Unix/Linux system

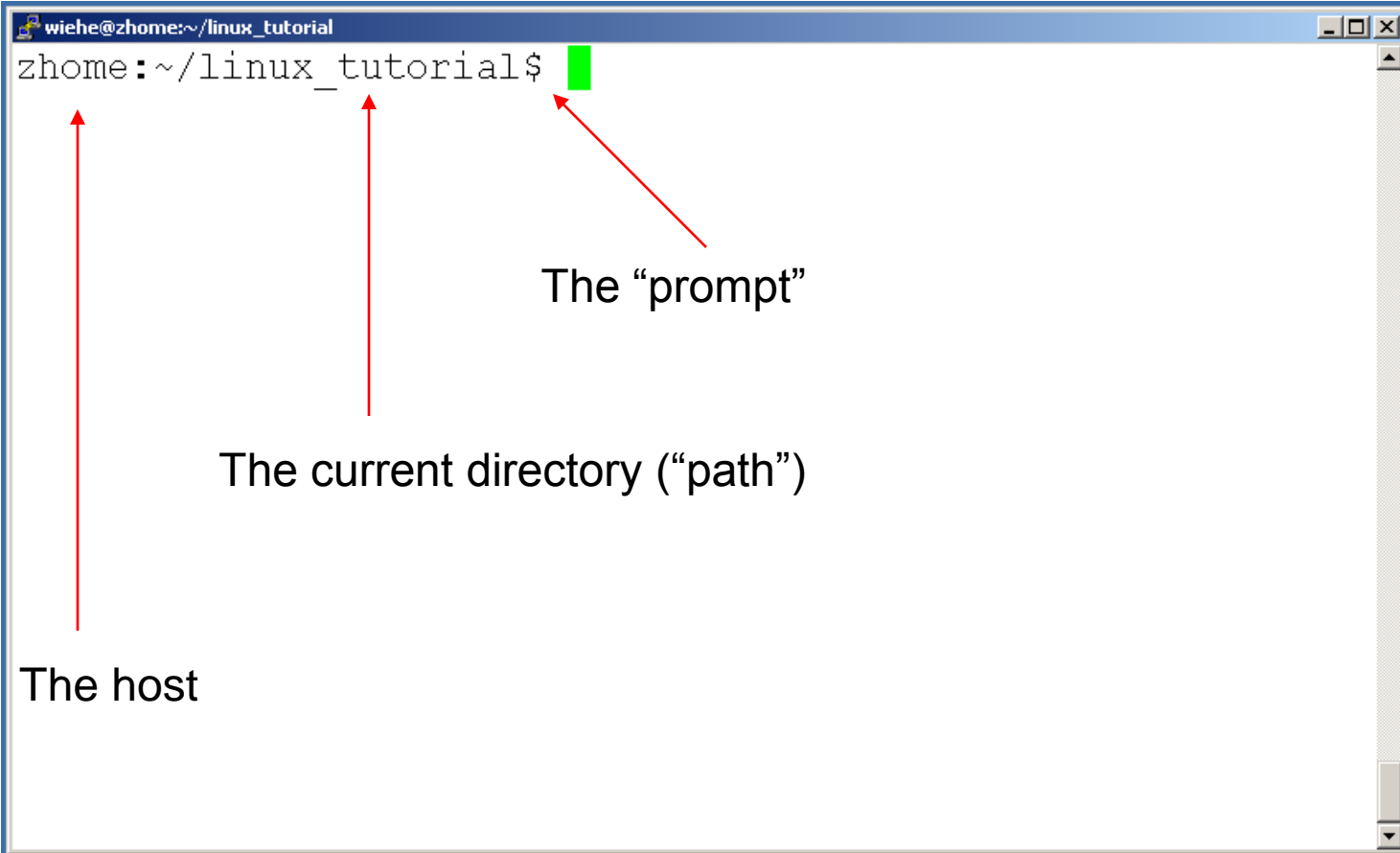
- ▶ Open up a terminal:

A screenshot of a terminal window. The title bar at the top reads "wiehe@zhome:~/linux_tutorial" and includes standard window control buttons (minimize, maximize, close). The terminal content shows the prompt "zhome:~/linux_tutorial\$" followed by a green cursor block. The window has a scroll bar 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: `wiehe@zhome:~/linux_tutorial` in the title bar and `zhome:~/linux_tutorial$` in the main area. A green cursor is positioned at the end of the prompt. Three red arrows point from text labels to parts of the prompt: one from 'The host' to 'wiehe', one from 'The current directory ("path")' to '~/linux_tutorial', and one from 'The "prompt"' to '\$'. The terminal window has standard window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

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

The host

The current directory ("path")

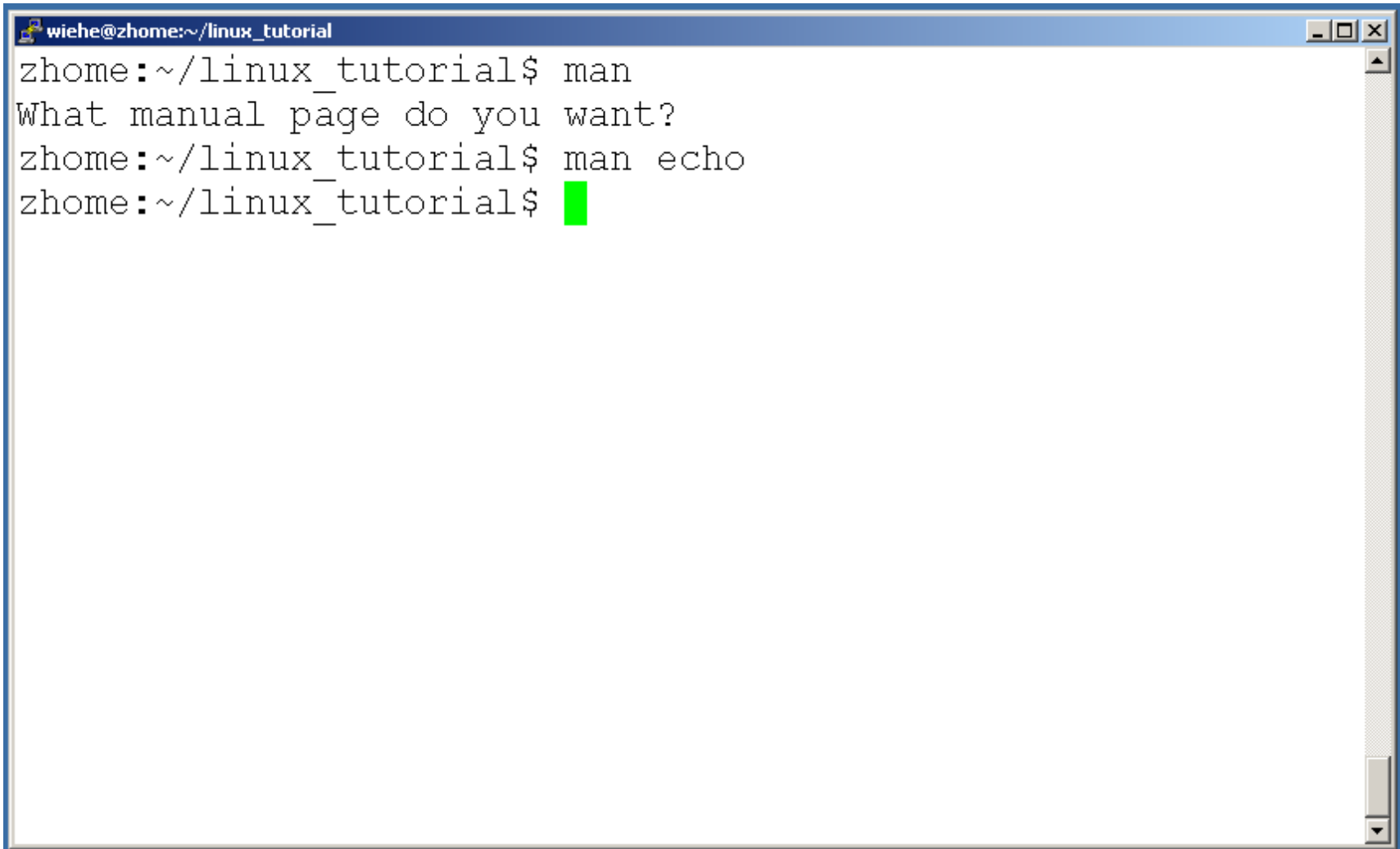
The "prompt"

Help!

- ▶ Endless number of commands/programs and parameters
- ▶ But: *“You never walk alone”*
- ▶ Whenever you need help with a command:
 - ▶ type “man” and the command name
 - ▶ run the command with `-h` or `-help` as parameter
 - ▶ ask google!!



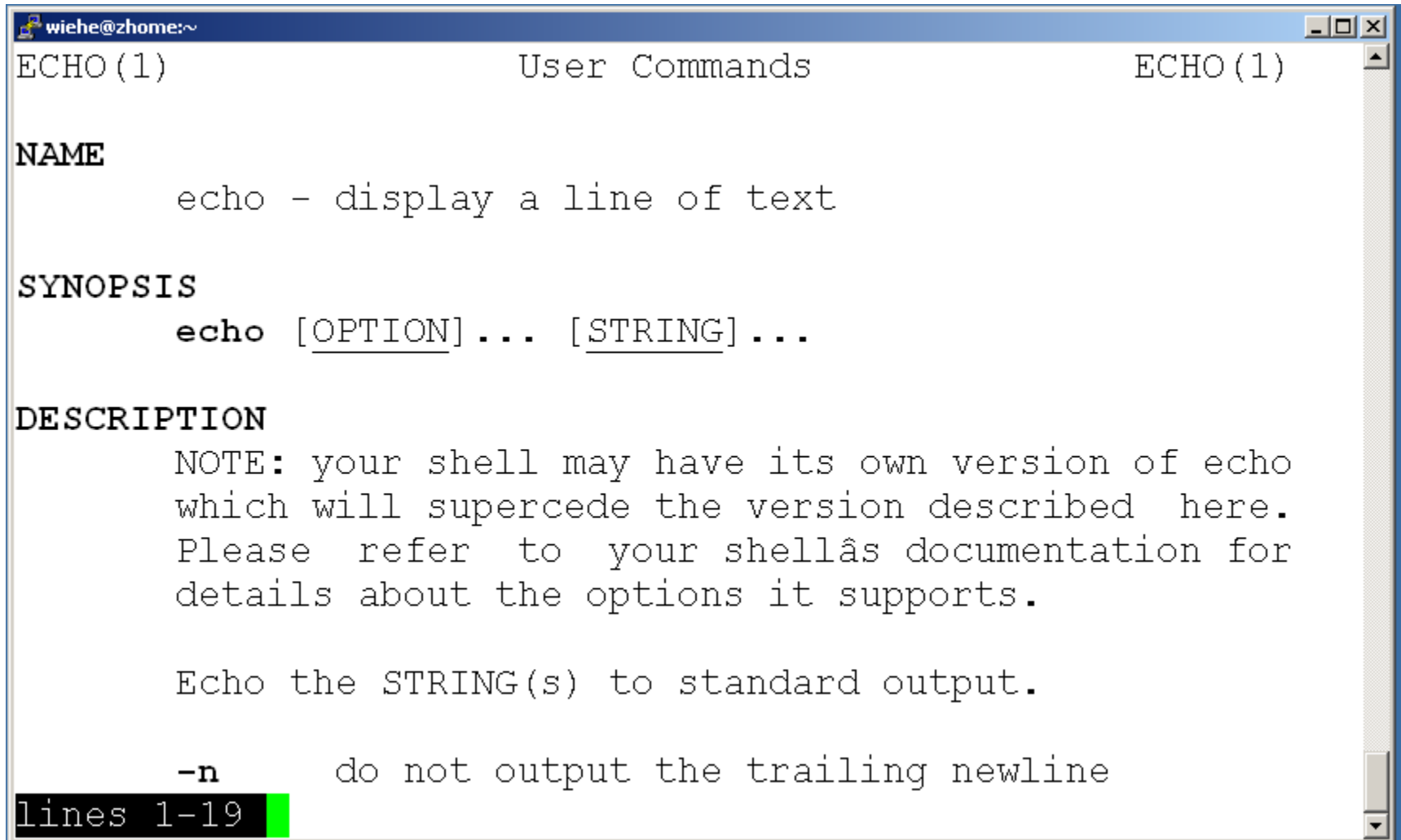
Help!



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

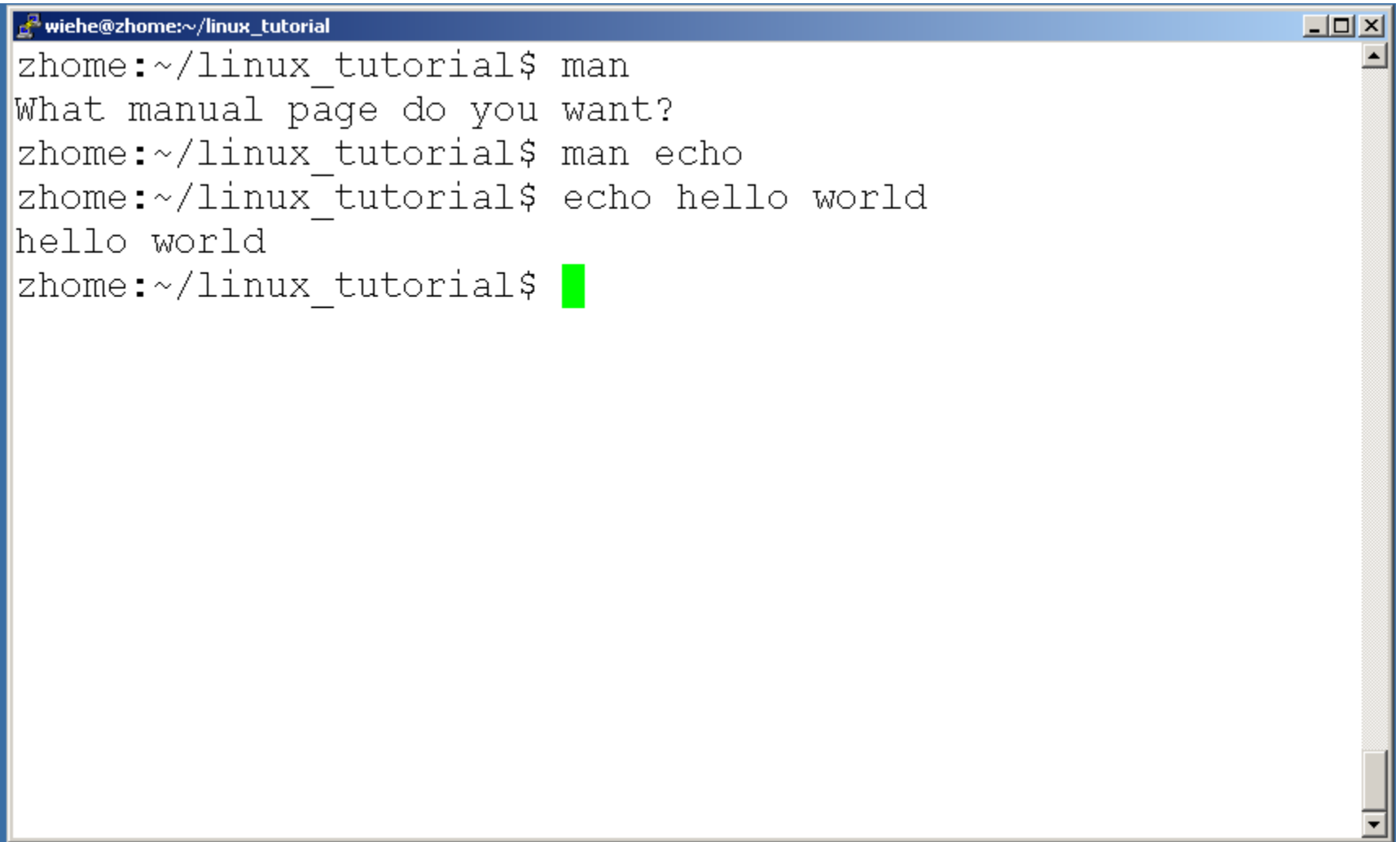


Help!



```
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!

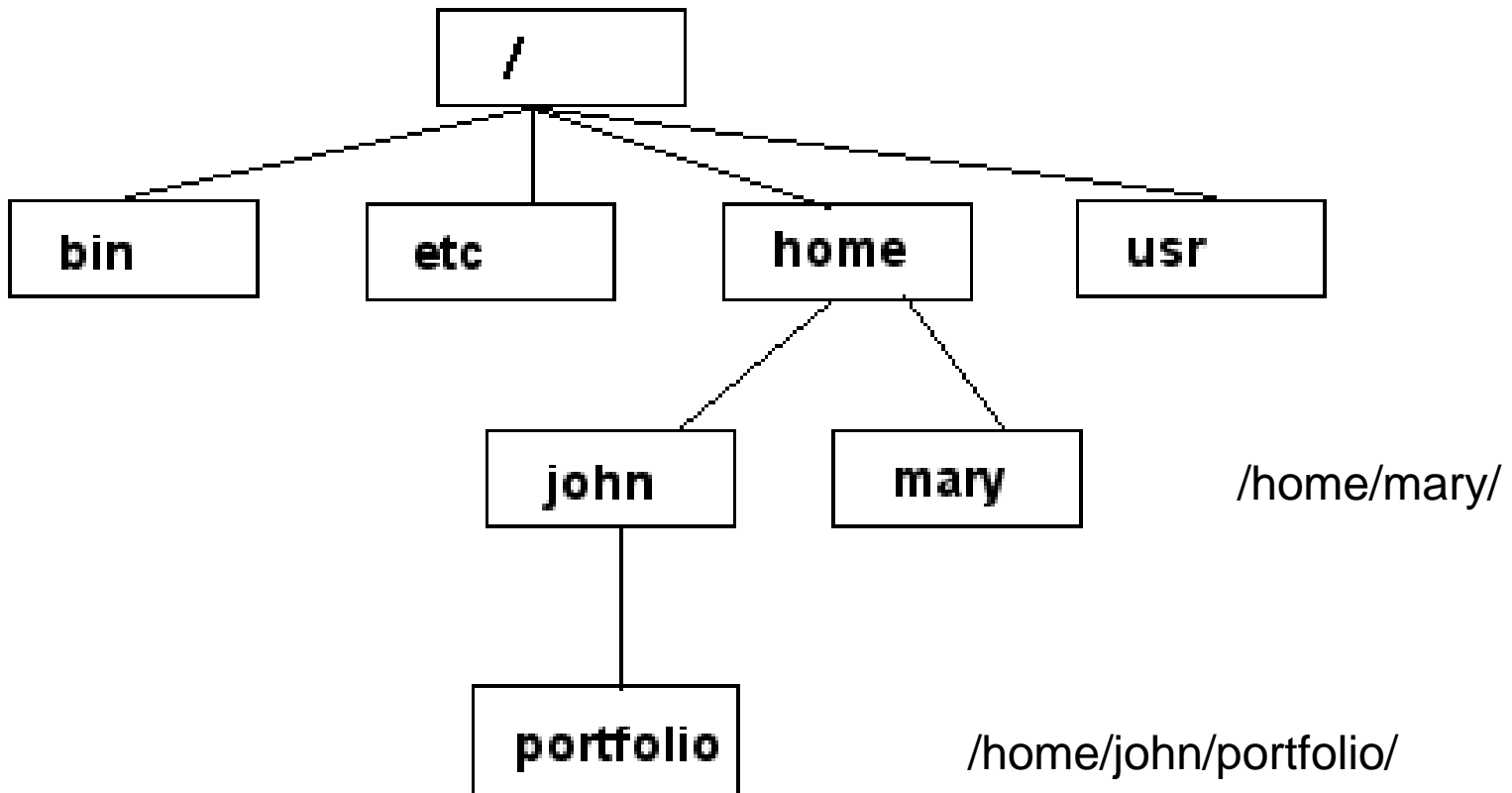


```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ man
What manual page do you want?
zhome:~/linux_tutorial$ man echo
zhome:~/linux_tutorial$ echo hello world
hello world
zhome:~/linux_tutorial$ █
```



Unix/Linux File System

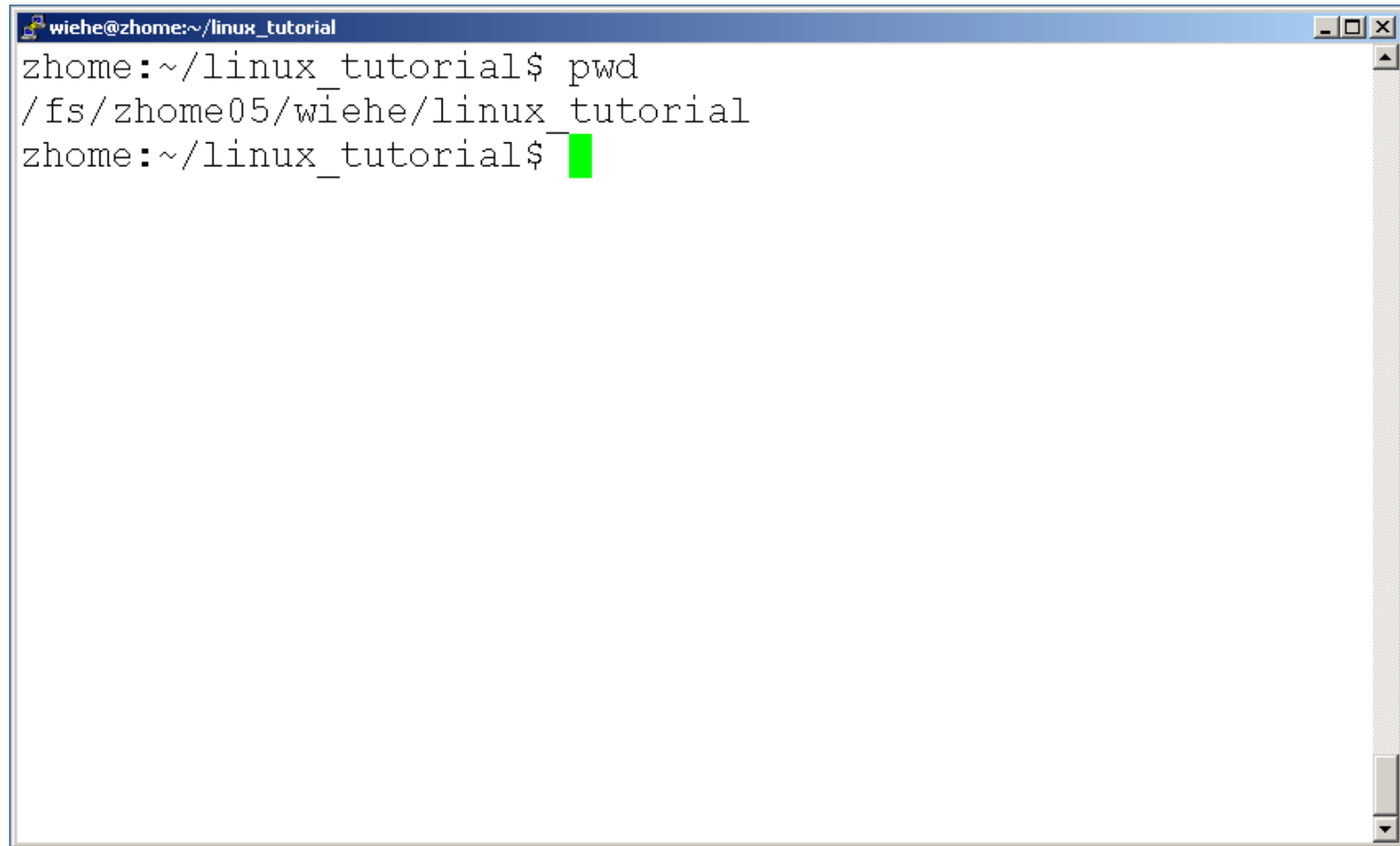
NOTE: Unix file names are **CASE SENSITIVE!**



The Path

Command: pwd

- ▶ To find your current path use “pwd”

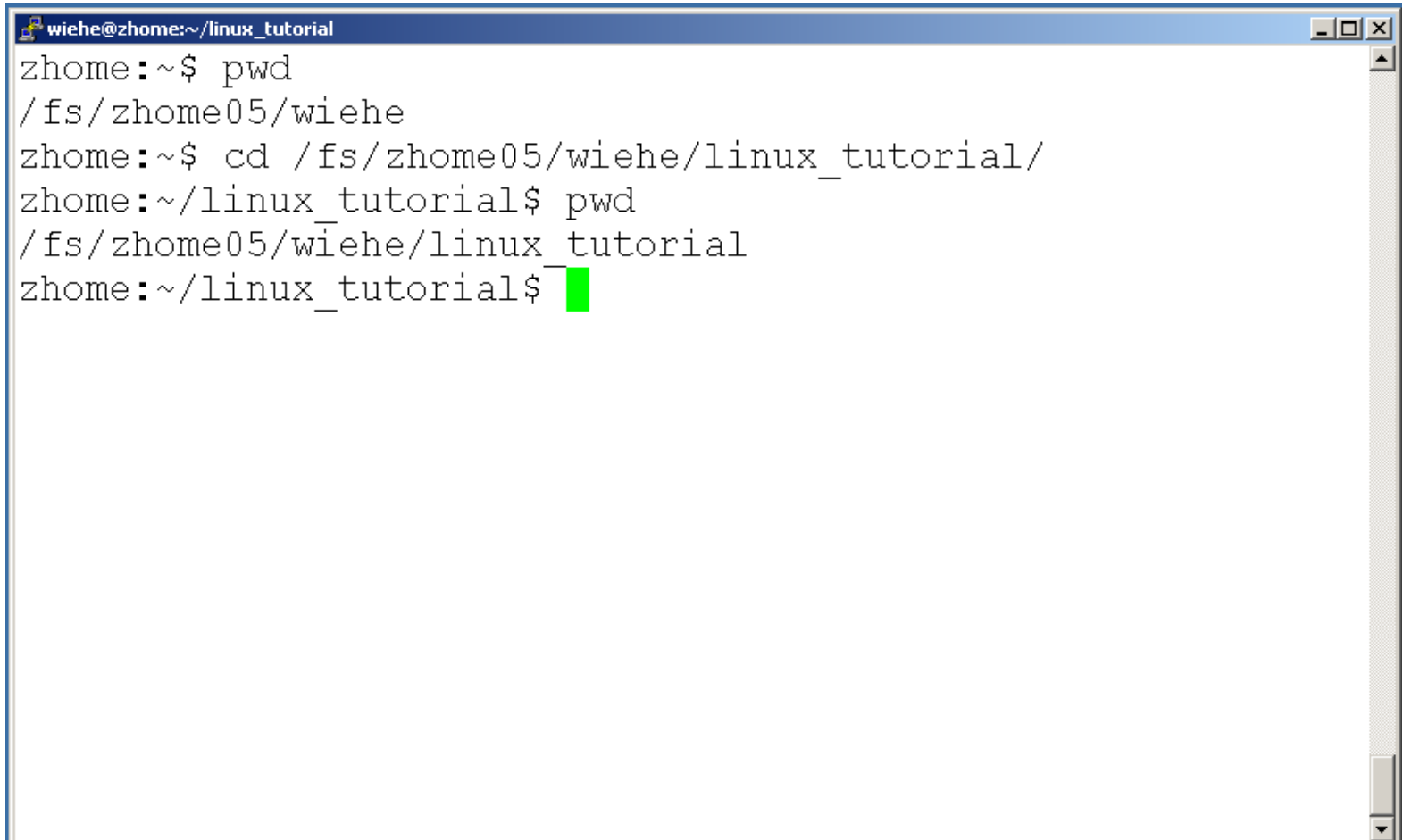


```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ pwd
/fs/zhome05/wiehe/linux_tutorial
zhome:~/linux_tutorial$
```

A terminal window with a blue title bar containing the text "wiehe@zhome:~/linux_tutorial". The window shows a shell prompt "zhome:~/linux_tutorial\$" followed by the command "pwd". The output of the command is "/fs/zhome05/wiehe/linux_tutorial". The prompt "zhome:~/linux_tutorial\$" is shown again on the next line, 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.

Command: cd

- ▶ To change to a specific directory use “cd”

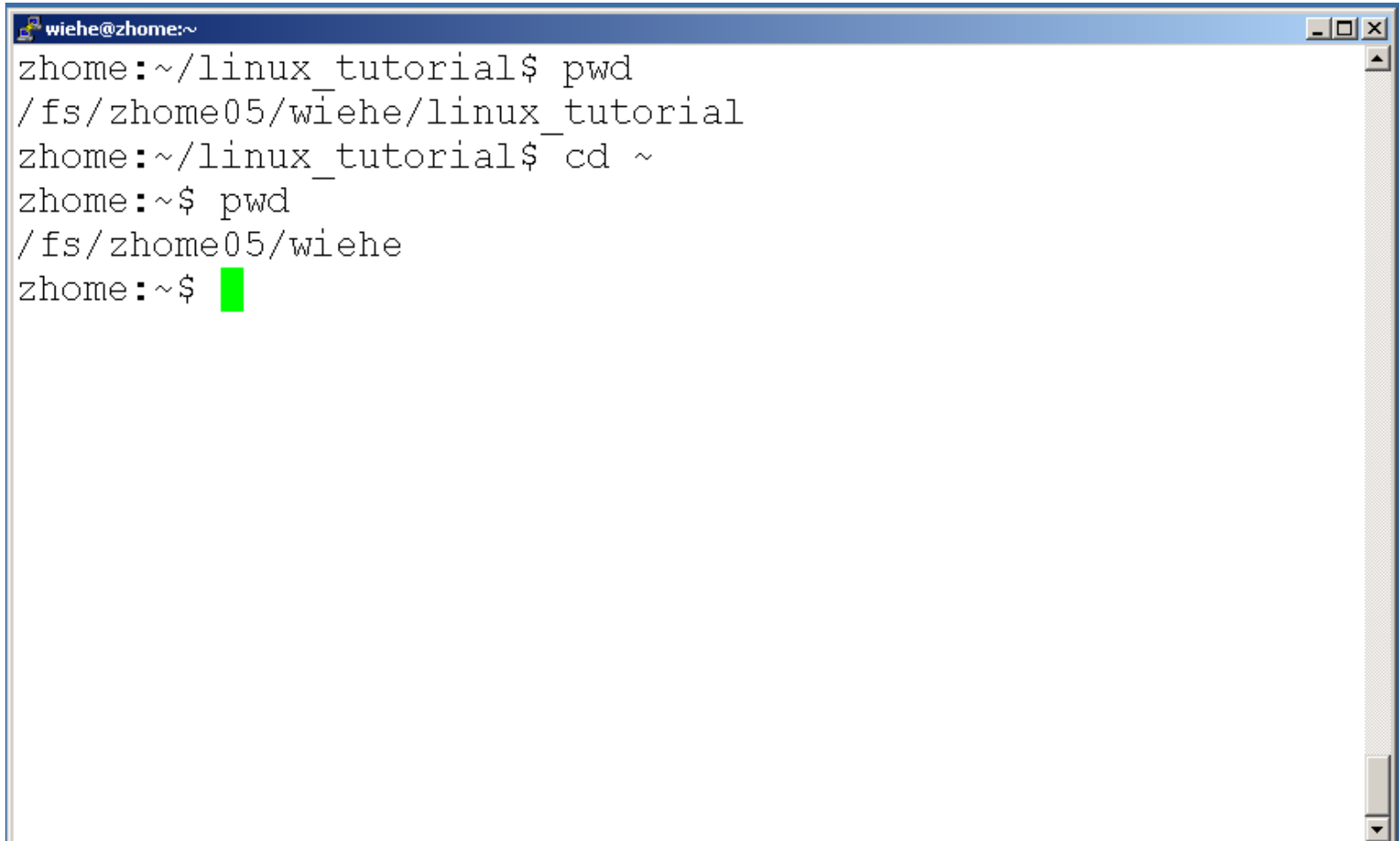
A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The terminal shows the following sequence of commands and outputs:

```
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$ █
```

The cursor is a green block at the end of the last line.

Command: cd

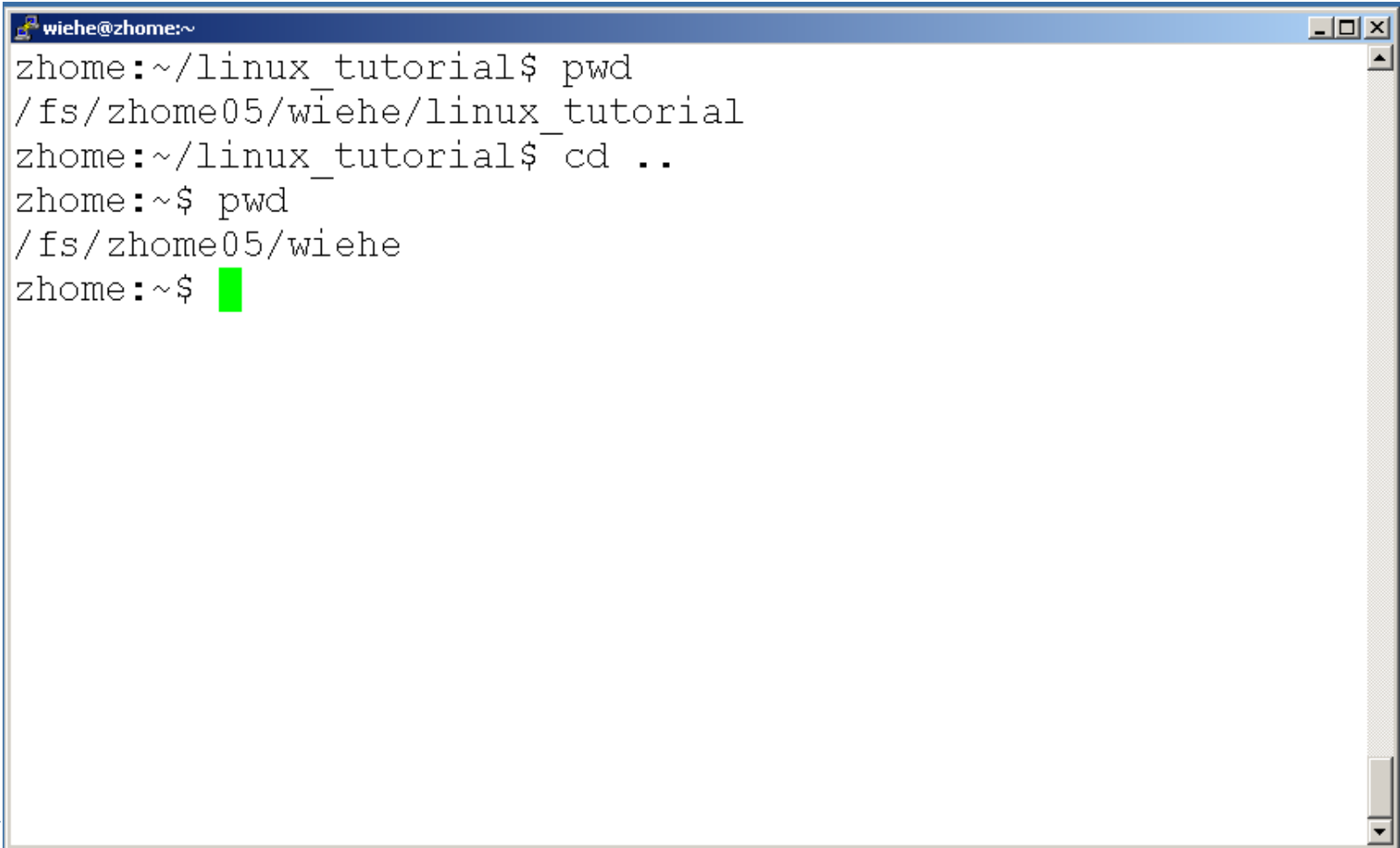
- ▶ “~” is the location of your home directory



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

Command: cd

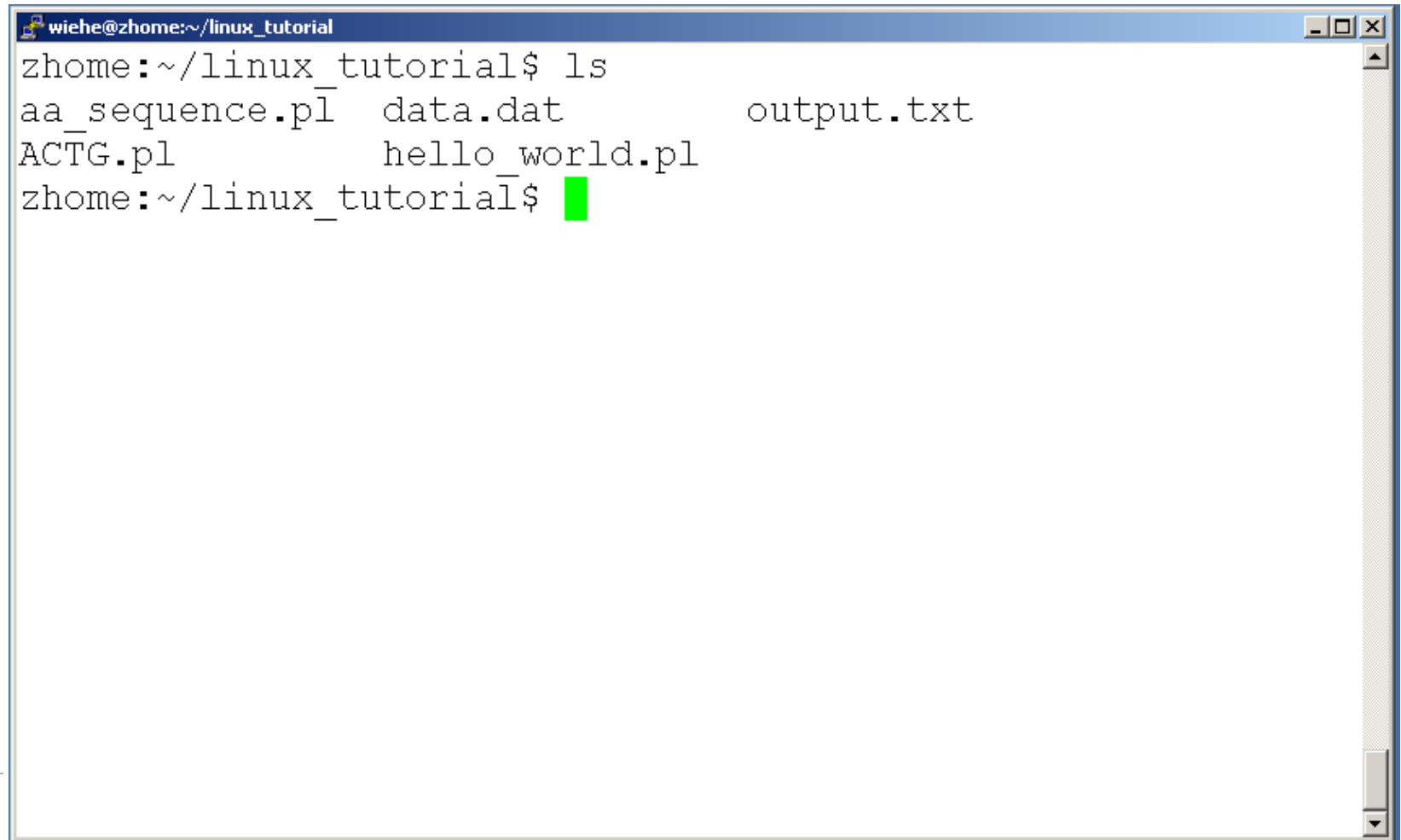
- ▶ “..” is the location of the directory below current one

A terminal window titled 'wiehe@zhome:~' showing a sequence of commands and their outputs. The user starts in the directory ~/linux_tutorial, runs 'pwd' to get /fs/zhome05/wiehe/linux_tutorial, then runs 'cd ..' to move to the parent directory. A subsequent 'pwd' command returns /fs/zhome05/wiehe. The prompt ends with a green cursor.

```
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”



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

A terminal window titled "wiehe@zhome:~/linux_tutorial" showing the execution of the "ls" command. The output lists four files: "aa_sequence.pl", "data.dat", "output.txt", and "ACTG.pl", with "hello_world.pl" listed on the same line as "ACTG.pl". The prompt "zhome:~/linux_tutorial\$" is shown again at the end of the output, followed by a green cursor block.

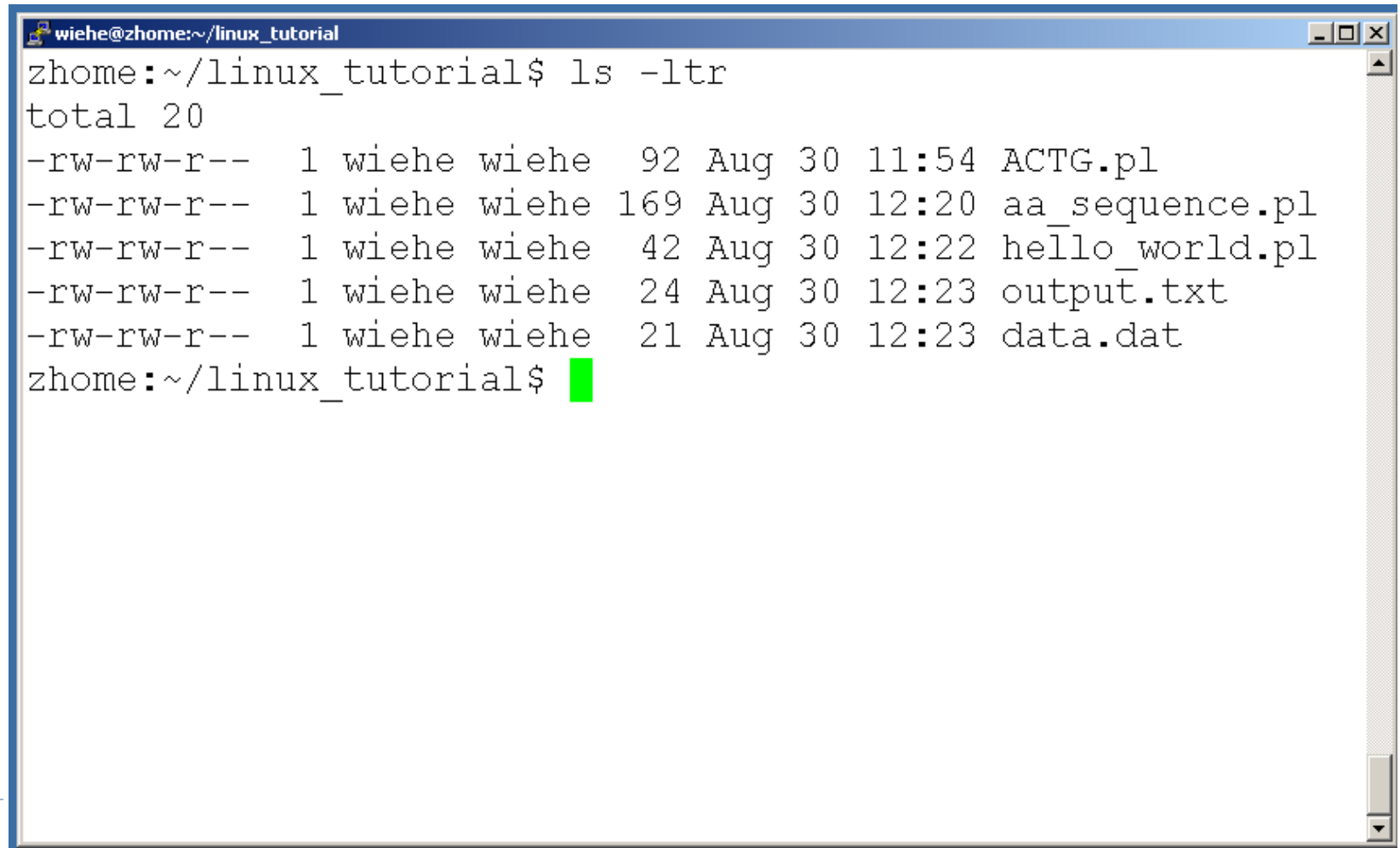
Command: ls

- ▶ **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

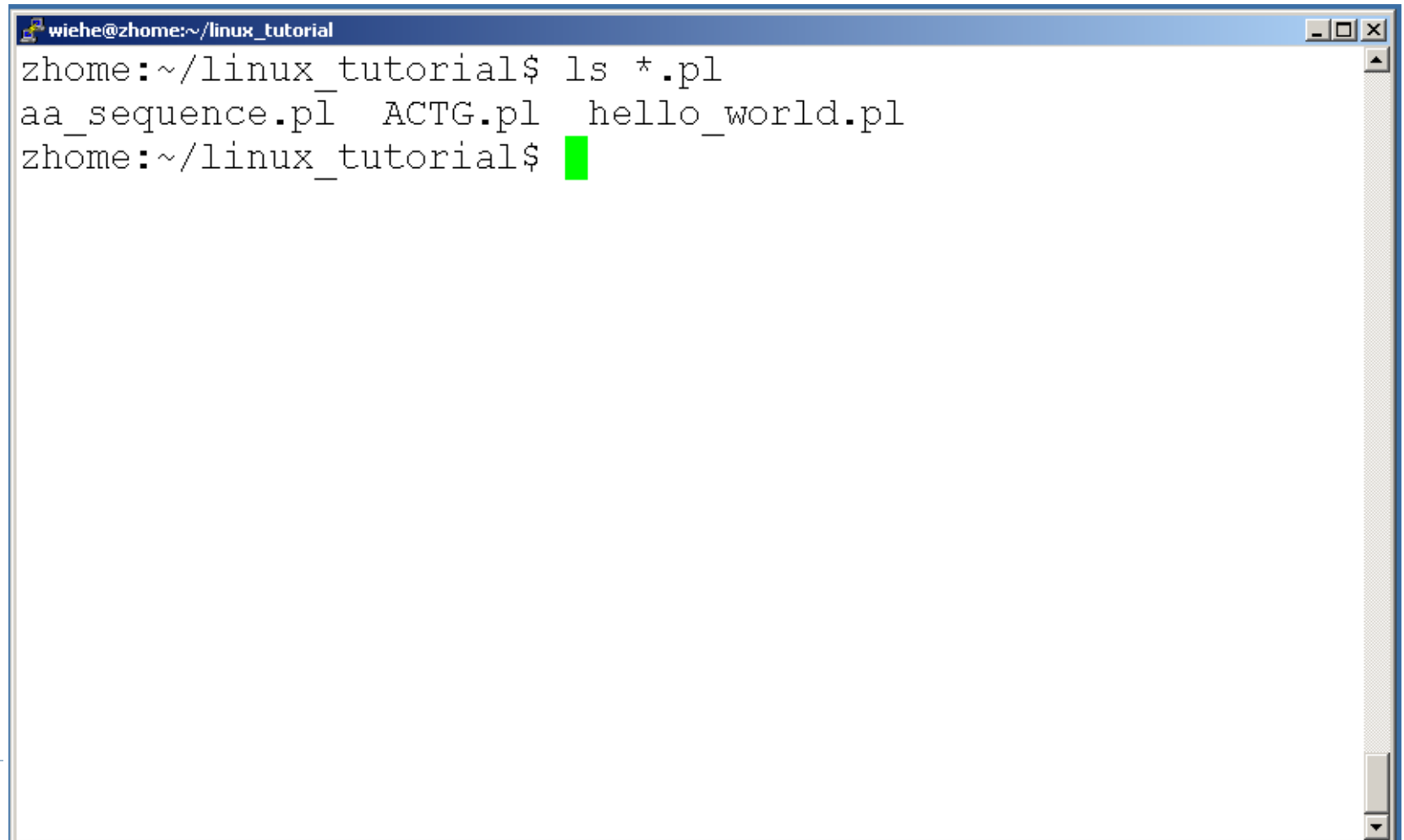
- ▶ List files by time in reverse order with long listing



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

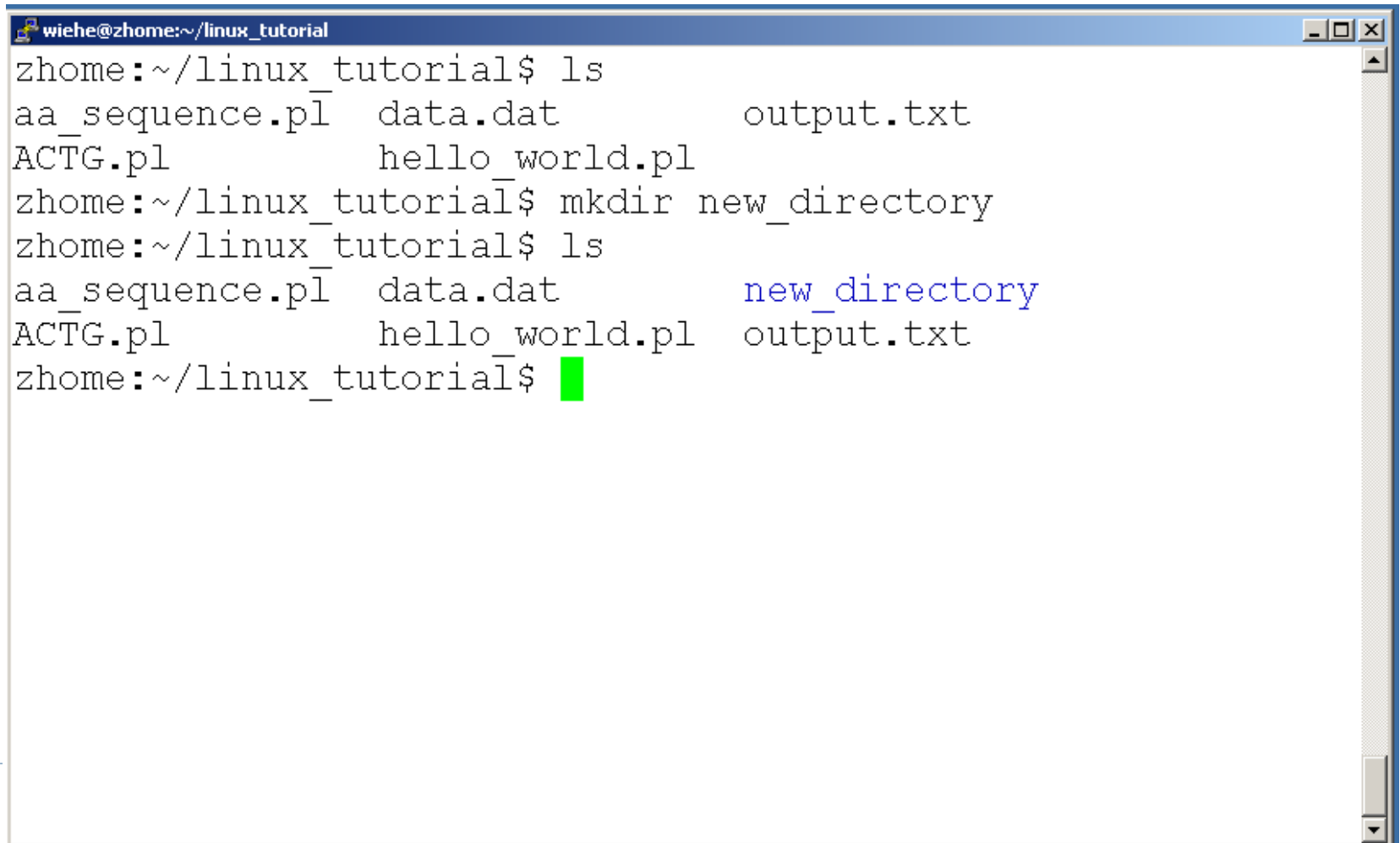
- ▶ “*” can be used as a wildcard in unix/linux



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


Command: mkdir

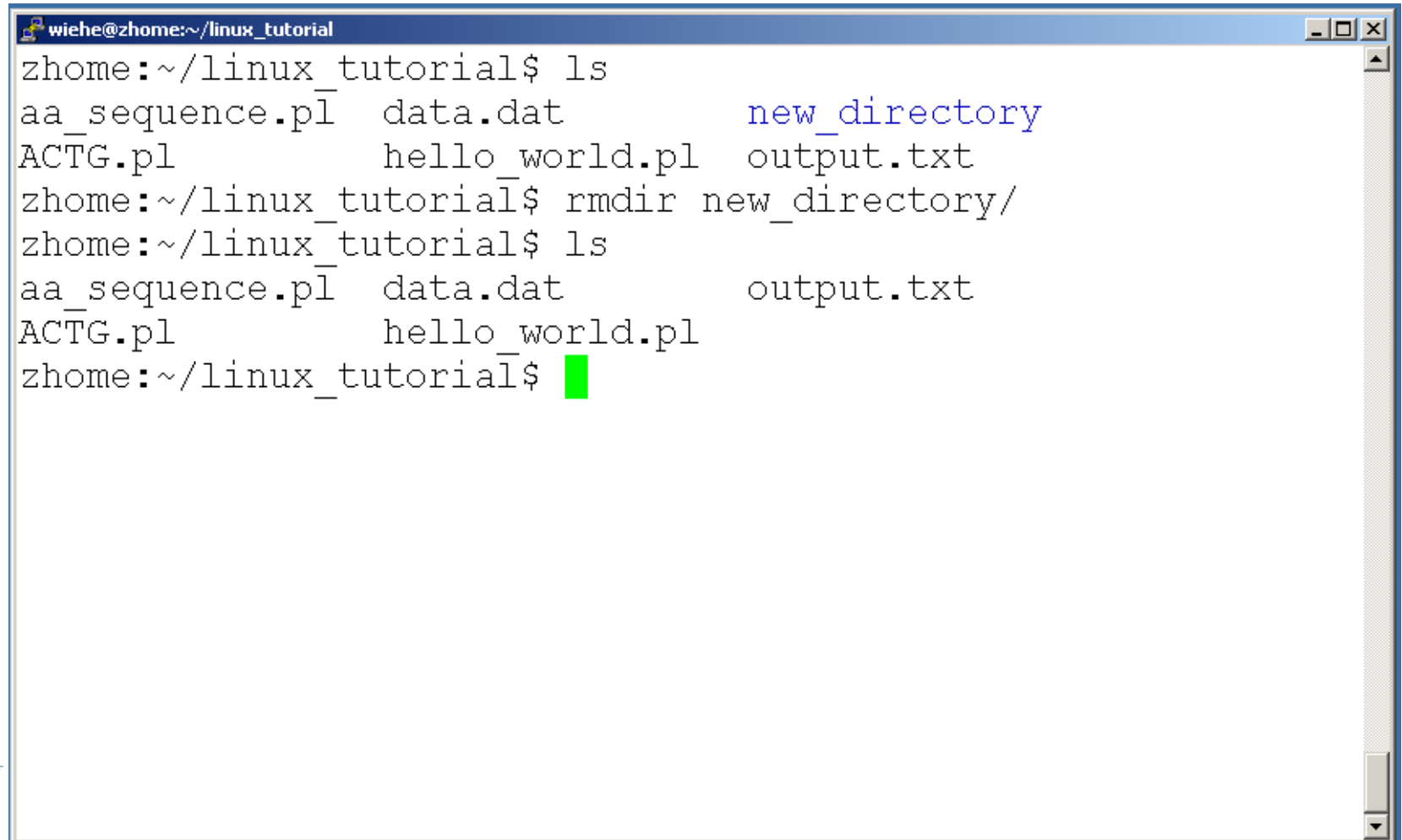
- ▶ To create a new directory use “mkdir”



```
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$ █
```

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”



Exercises 1-5



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 and concatenating files

- ▶ Example: Concatenate three FASTQ files

```
cat SRA0001.fq SRA0002.fq SRA0003.fq > SRA000_all.fq
```



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

- ▶ **Example: Check a SAM file**

```
less -S mapped_reads.sam
```



Command: head

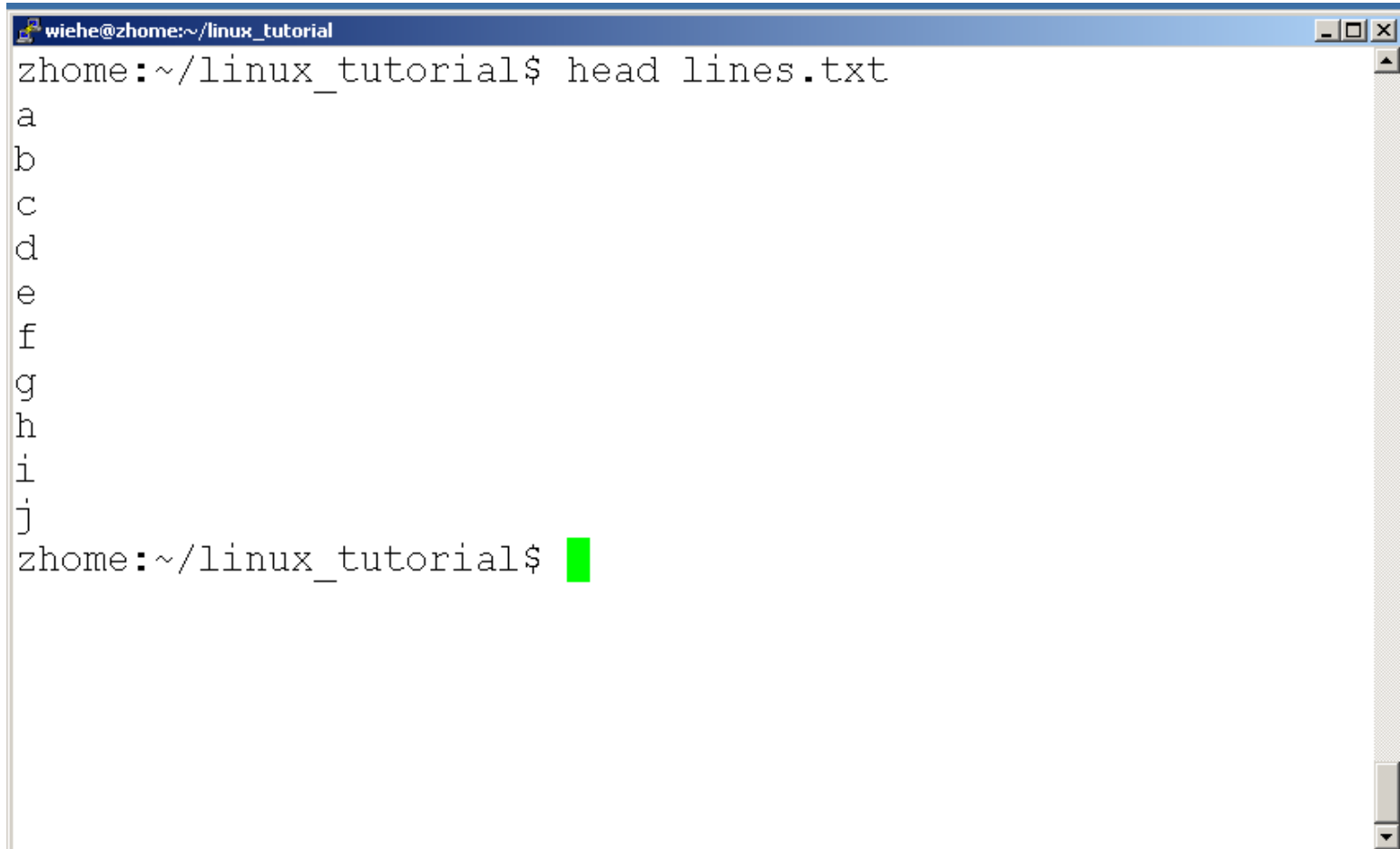
- ▶ “head” displays the top part of a file
 - ▶ By default it shows the first 10 lines
 - ▶ -n option allows you to change that
-
- ▶ **Example: Display the first 10 reads of your dataset**

```
head -n 40 SR012310.fq
```



Command: head

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

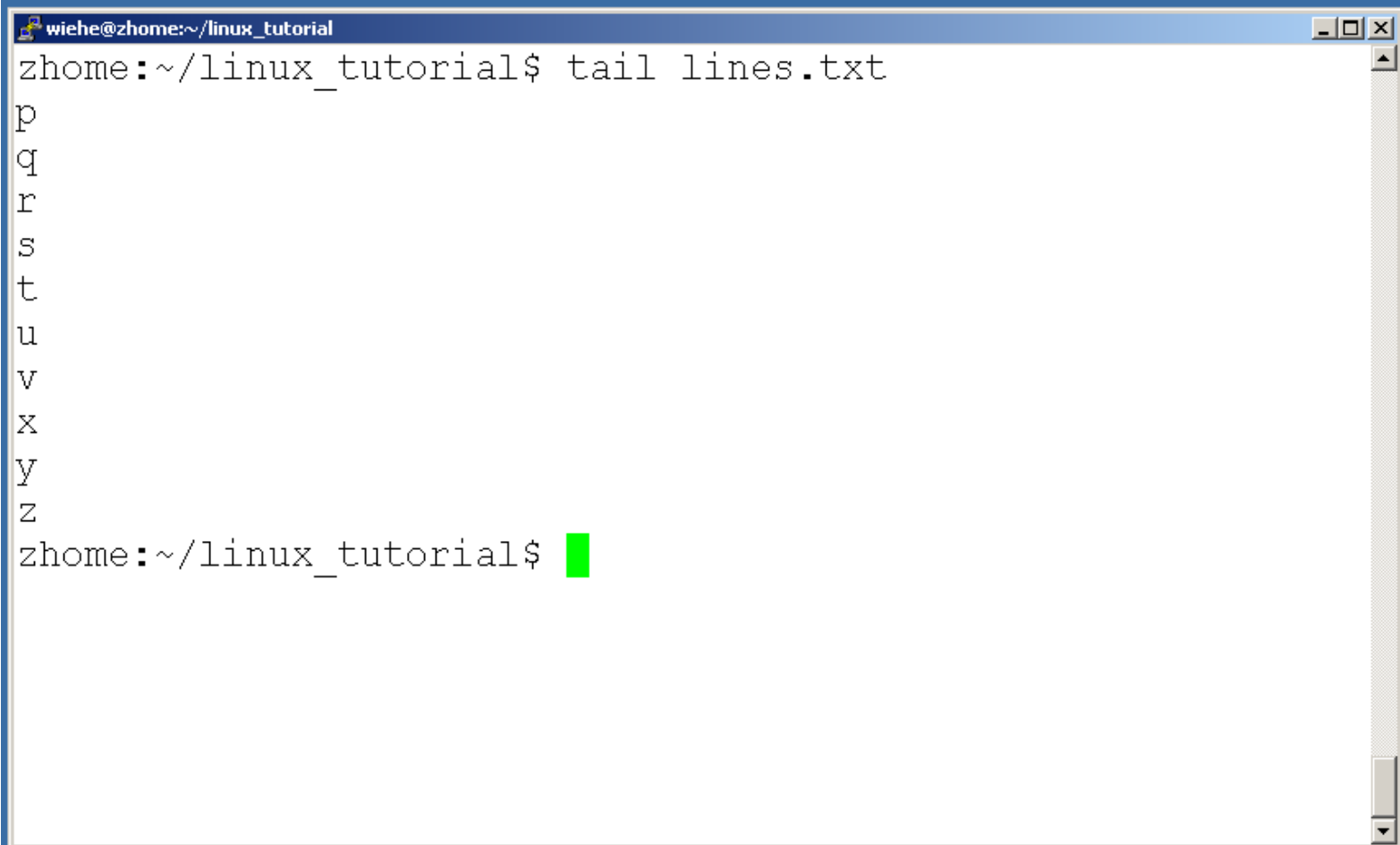
A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The terminal content shows the command 'head lines.txt' being executed, resulting in the output 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'. The prompt 'zhome:~/linux_tutorial\$' is visible at the end of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



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

The image shows a terminal window with a blue title bar. The title bar text is "wiehe@zhome:~/linux_tutorial". The terminal content shows the command "tail lines.txt" being executed, which outputs the last lines of the file "lines.txt": "p", "q", "r", "s", "t", "u", "v", "x", "y", and "z". The prompt "zhome:~/linux_tutorial\$" is shown again at the bottom with a green cursor block.



Creating files in Unix/Linux

- ▶ Requires the use of an Editor
- ▶ Various Editors:
 - 1) nano / pico
 - 2) vi
 - 3) emacs



Editing a file using pico or nano

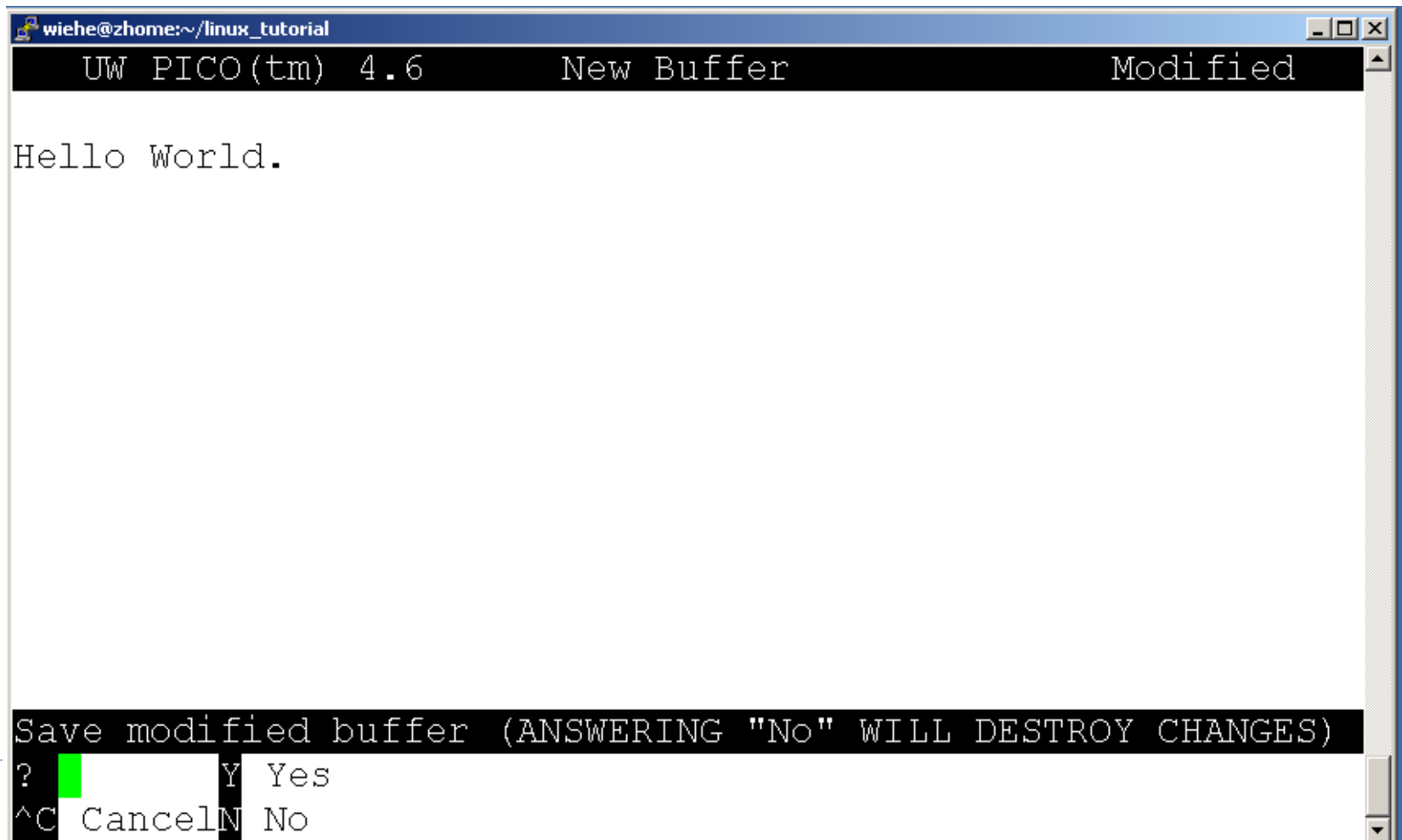
- ▶ Type “pico” or “nano” at the prompt

```
wiehe@zhome:~/linux_tutorial
UW PICO (tm) 4.6      New Buffer

^G Get He ^O WriteO ^R Read F ^Y Prev P ^K Cut Te ^C Cur Po
^X Exit   ^J Justif ^W Where ^V Next P ^U UnCut ^T To Spe
```

Editing a file using pico

- ▶ To save use “ctrl-x”



The screenshot shows a terminal window with the pico text editor. The title bar reads "wiehe@zhome:~/linux_tutorial". The editor's status bar at the top displays "UW PICO (tm) 4.6", "New Buffer", and "Modified". The main editing area contains the text "Hello World.". At the bottom, a black prompt bar asks "Save modified buffer (ANSWERING 'No' WILL DESTROY CHANGES)". Below this, a green cursor is positioned over the letter 'Y' in the "Y Yes" option. The other options are "?", "^C Cancel", and "N No".

```
wiehe@zhome:~/linux_tutorial
UW PICO (tm) 4.6          New Buffer          Modified
Hello World.

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES)
? Y Yes
^C Cancel N No
```

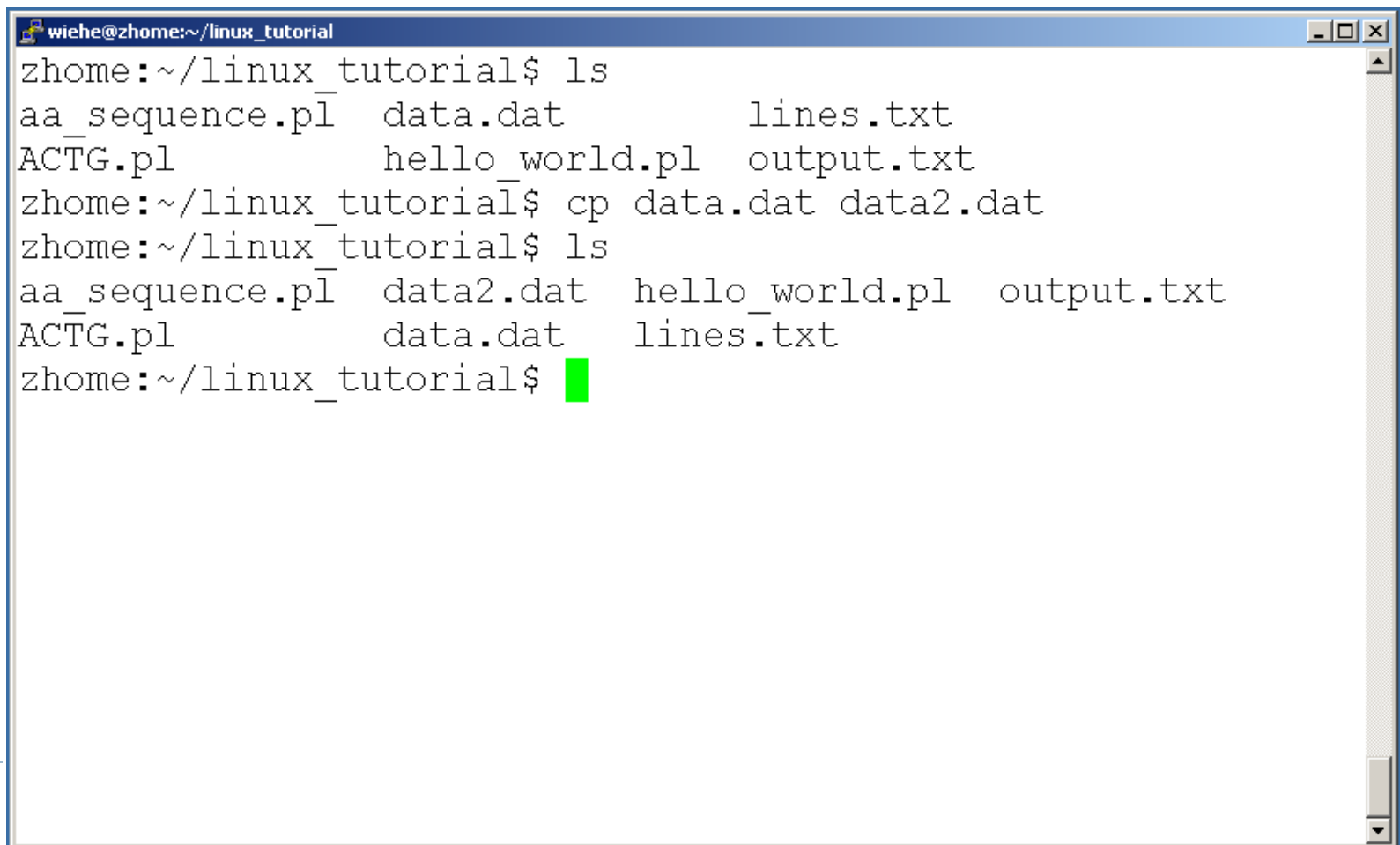
File Commands

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



Command: cp

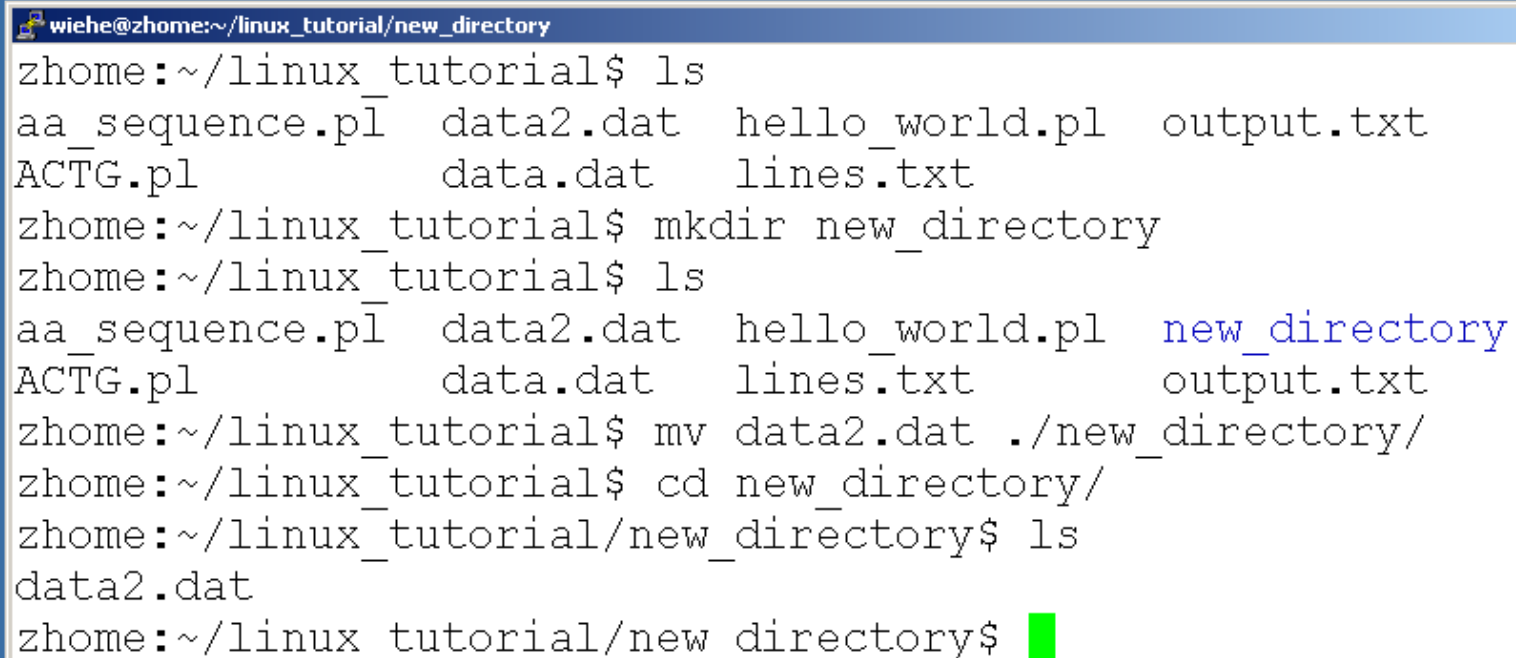
- ▶ To copy a file use “cp”



```
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

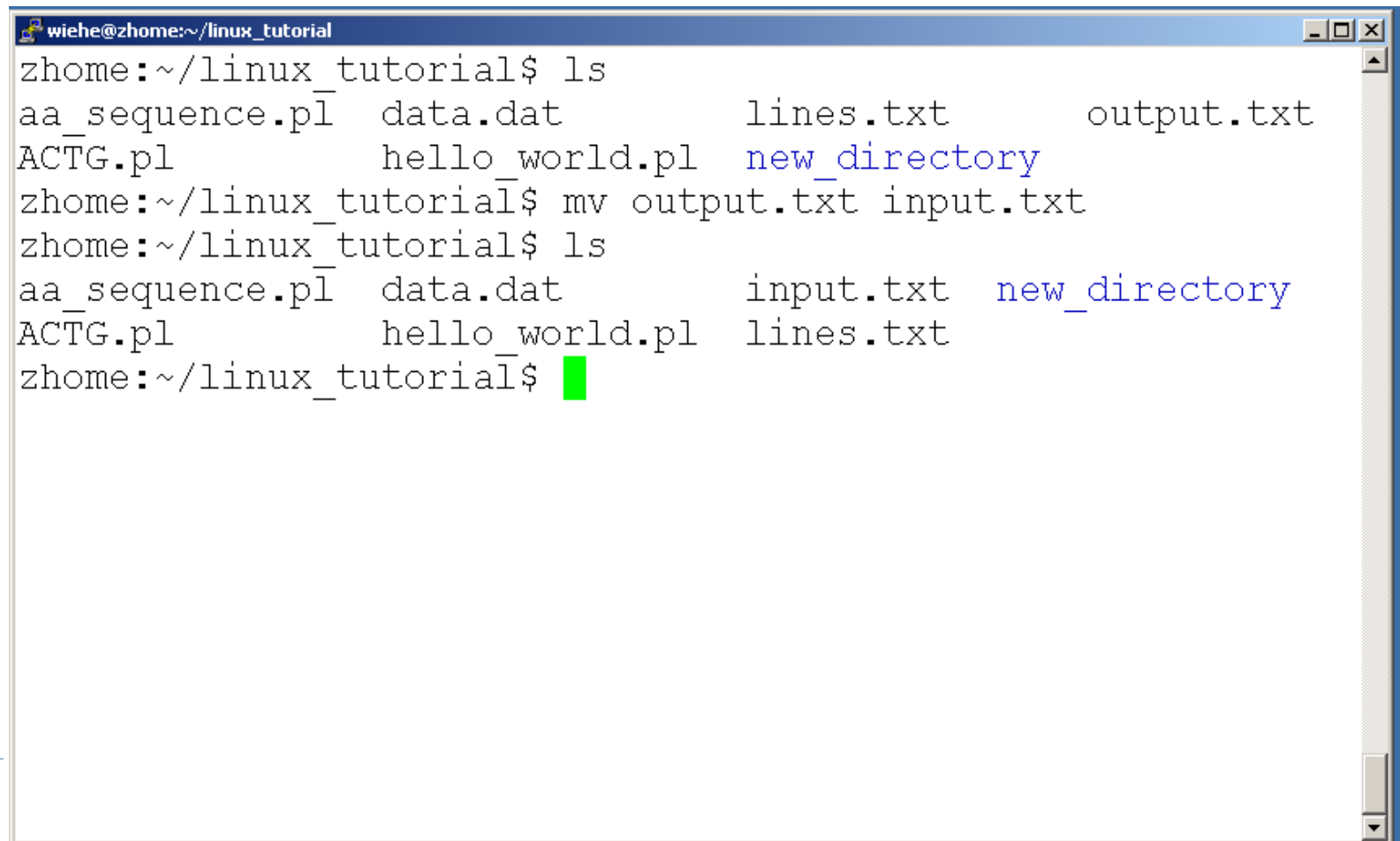
- ▶ To move a file to a different location use “mv”



```
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

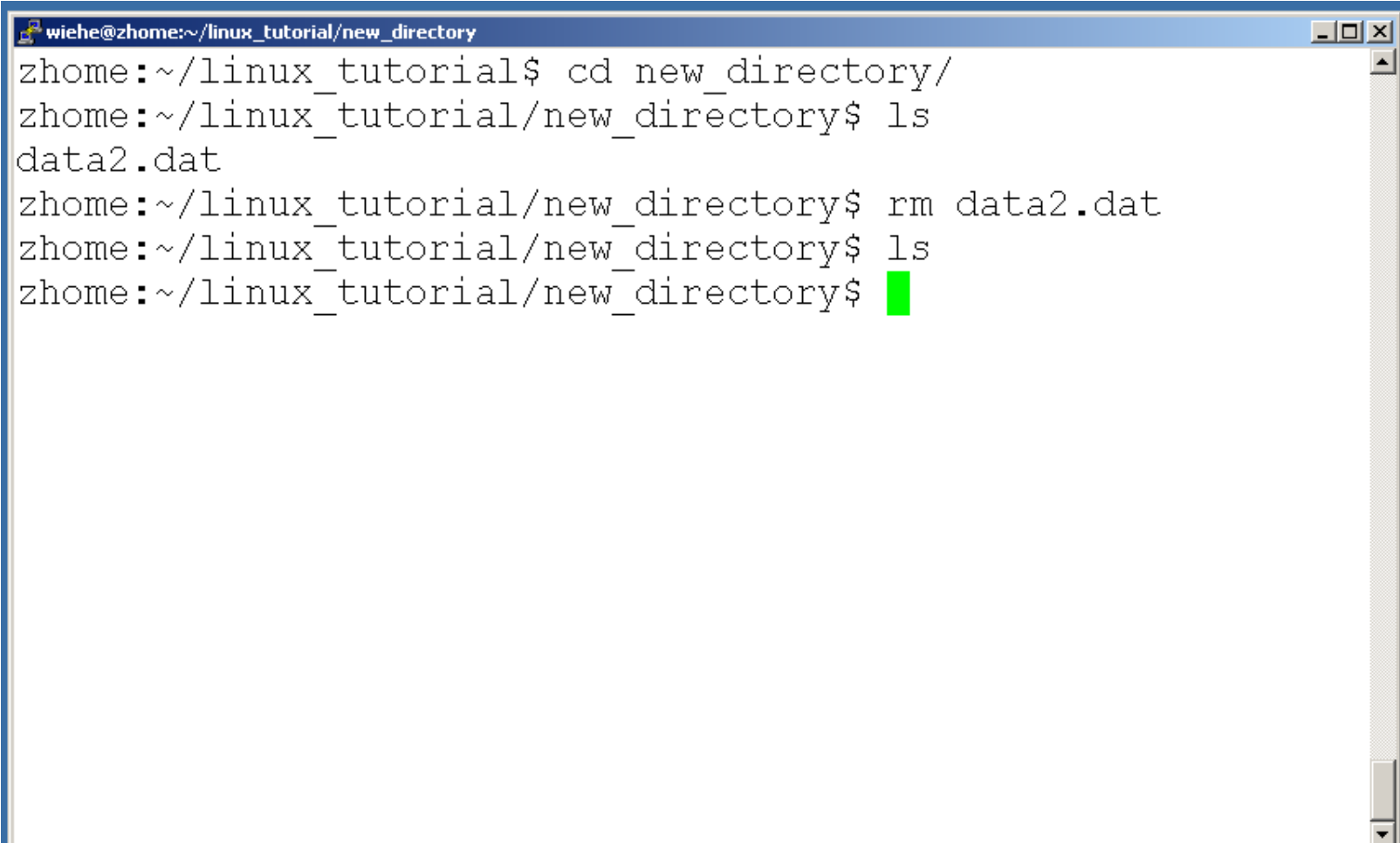
- ▶ mv can also be used to rename a file



```
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$ █
```

The image shows a terminal window with a blue title bar containing the text "wiehe@zhome:~/linux_tutorial/new_directory". The terminal content shows a sequence of commands and their outputs: "cd new_directory/" changes the current directory; "ls" lists the contents, showing "data2.dat"; "rm data2.dat" removes the file; and a second "ls" command is executed, with a green cursor block appearing at the end of the line, indicating the terminal is ready for the next command.

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



Exercises 6-10



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

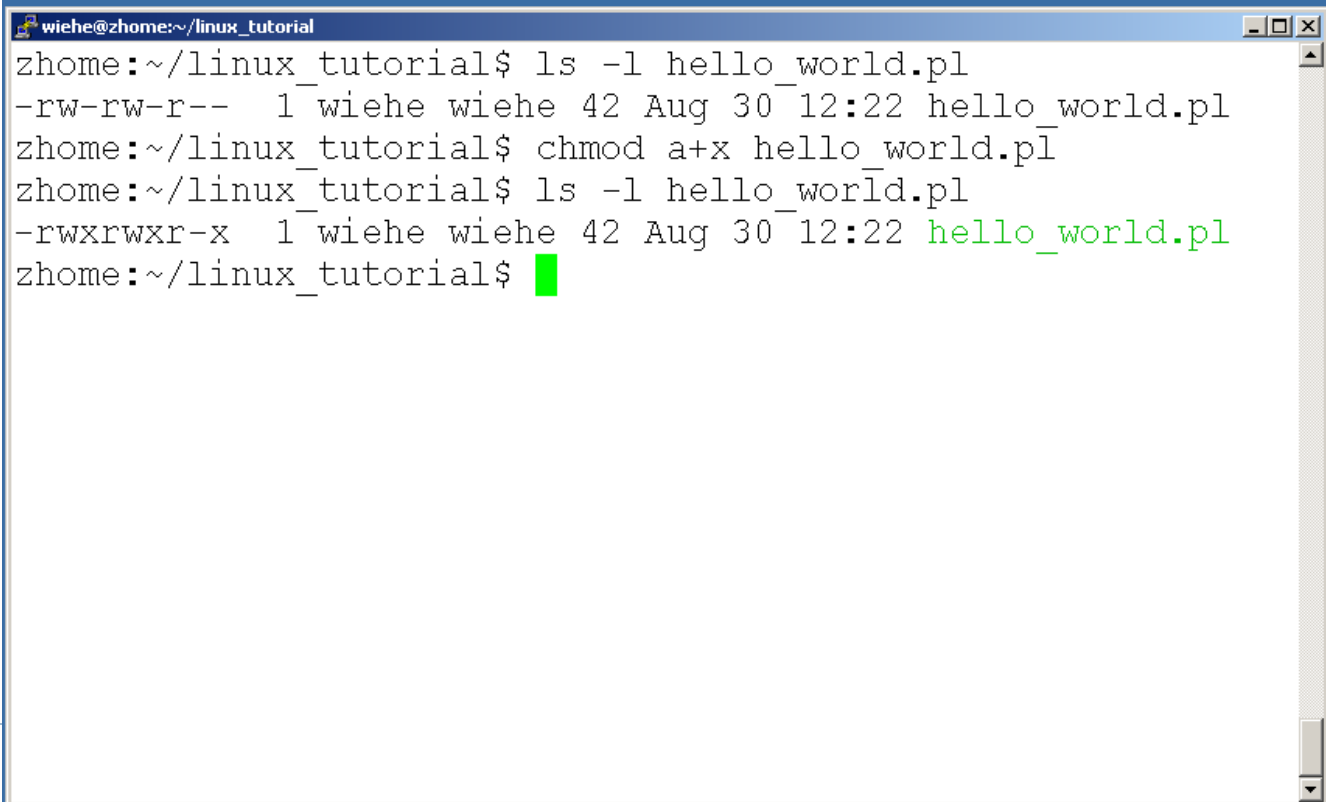
File Permissions

```
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 its 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 the execution of the chmod command. The window has a blue title bar and standard window controls. The terminal text is as follows:

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

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

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

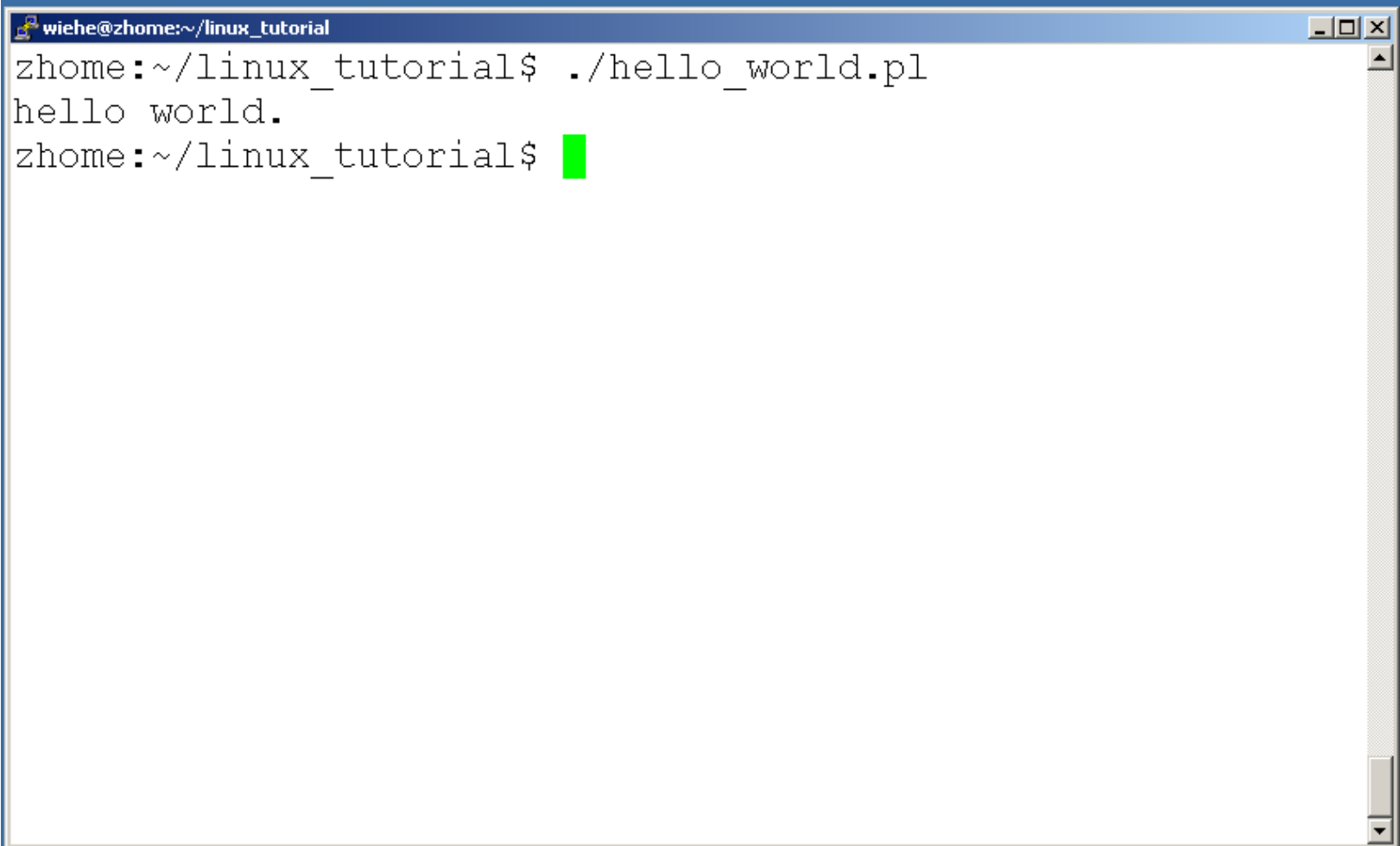
- ▶ **Example:**

```
wget http://bwa.tar.gz  
tar xvfz bwa.tar.gz  
chmod u+x bwa  
./bwa
```



Running a program: an example

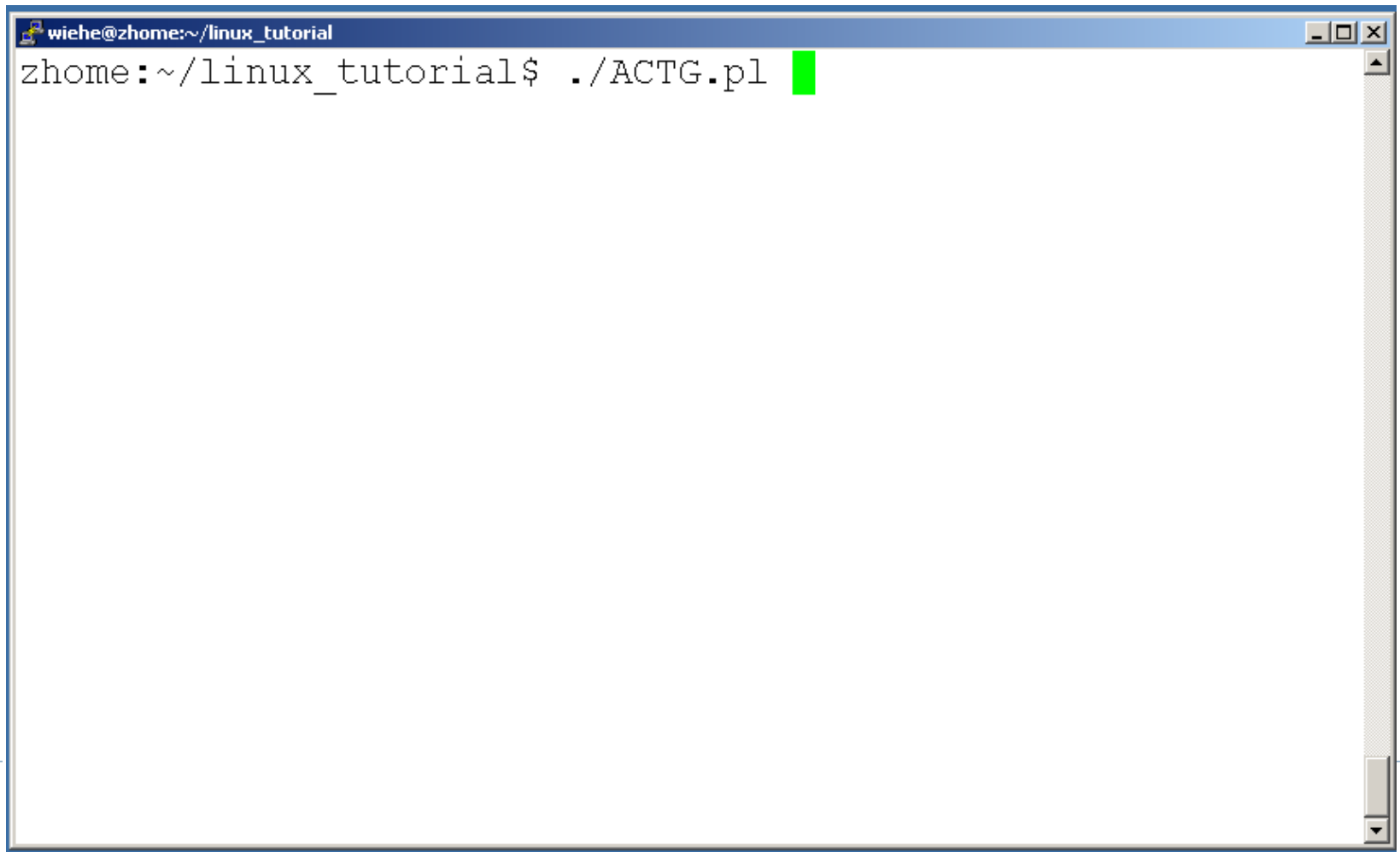
- ▶ Running the sample perl script “hello_world.pl”

A terminal window with a blue title bar containing the text "wiehe@zhome:~/linux_tutorial". The terminal content shows a shell prompt "zhome:~/linux_tutorial\$" followed by the command "./hello_world.pl". The output is "hello world." on the next line. A second shell 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 vertical 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 corner and a vertical scrollbar on the right side.

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

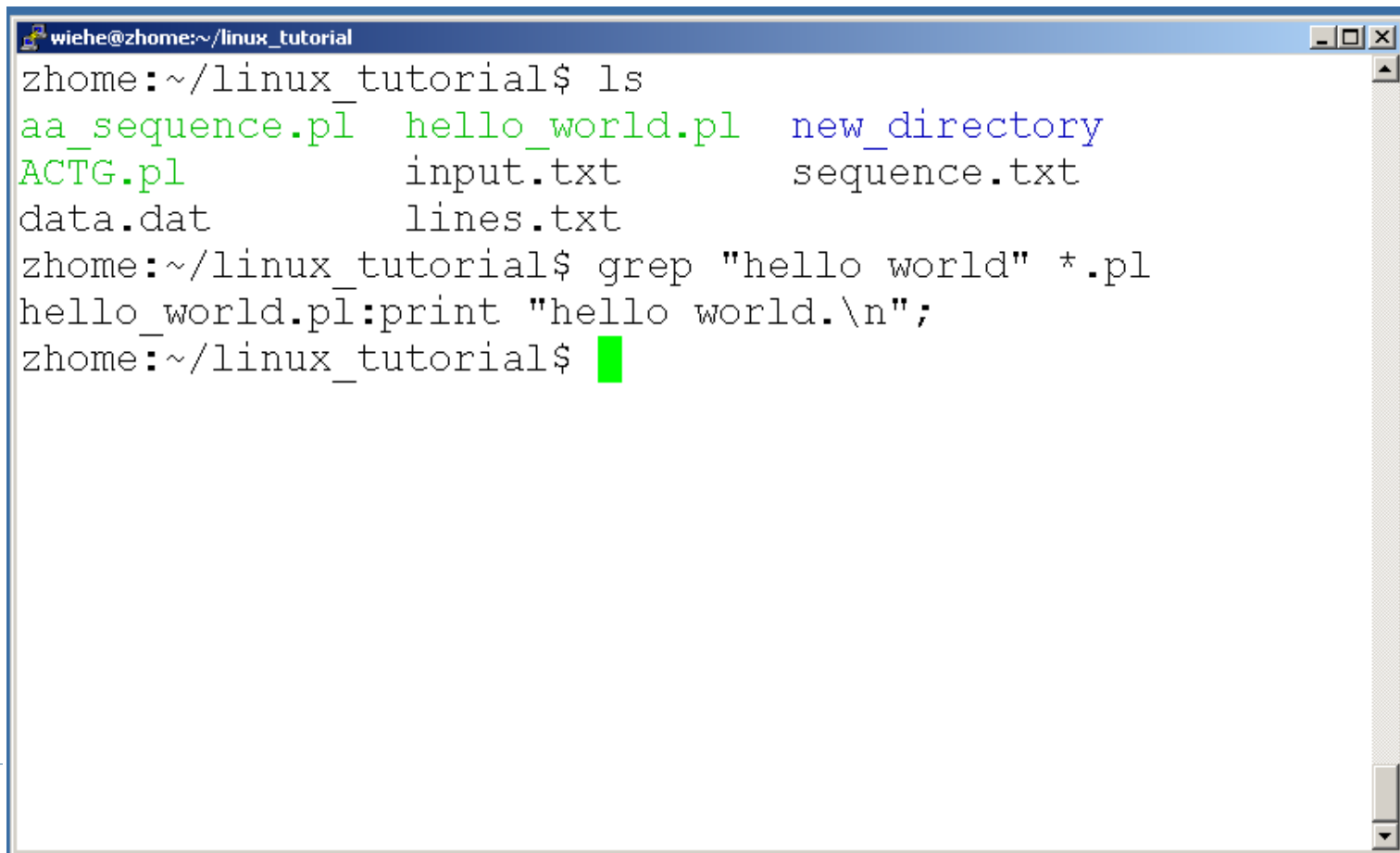
Command: wc

- ▶ 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



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$
```

Exercises 11-15



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**

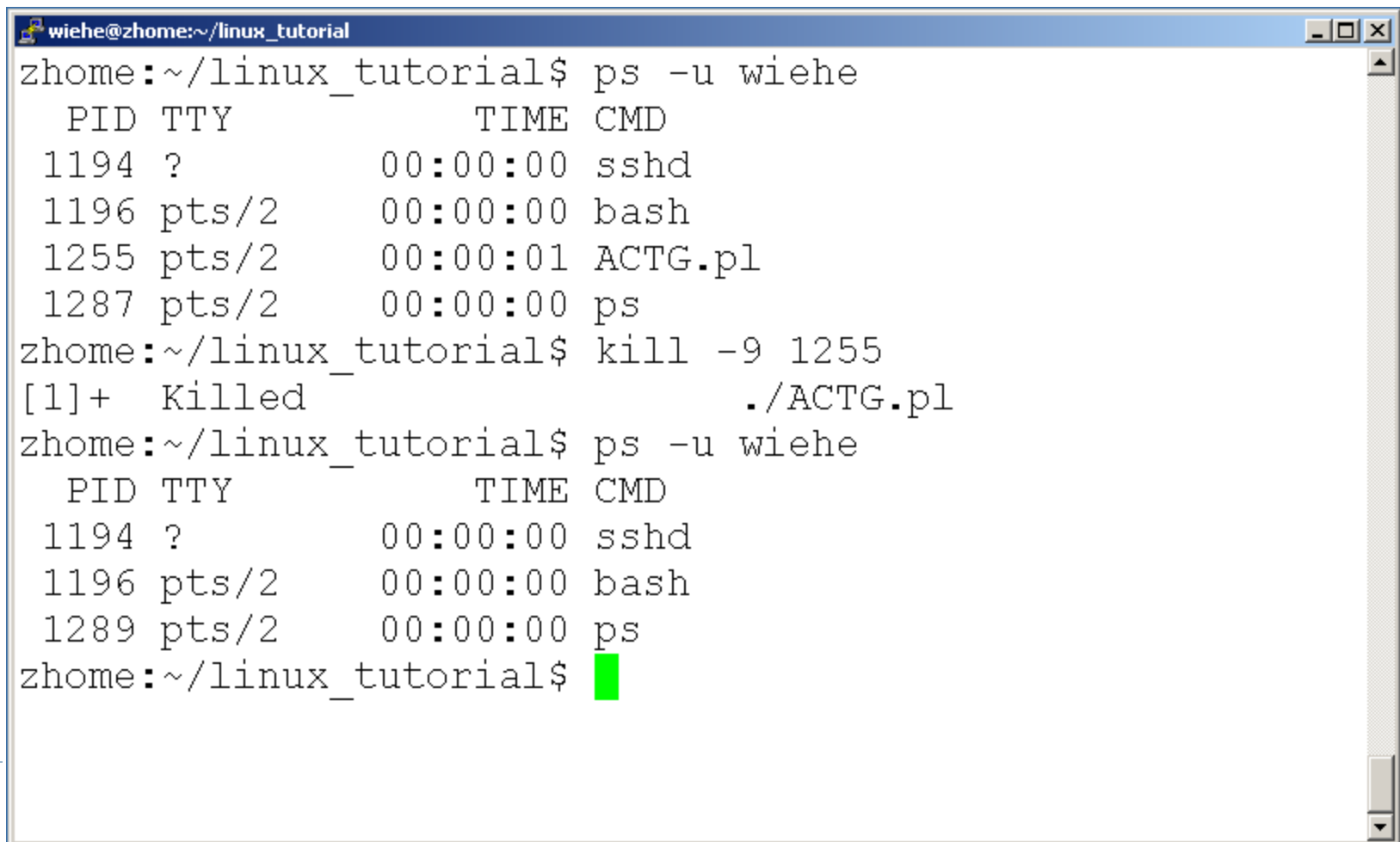
Unix Web Resources

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



Command: kill

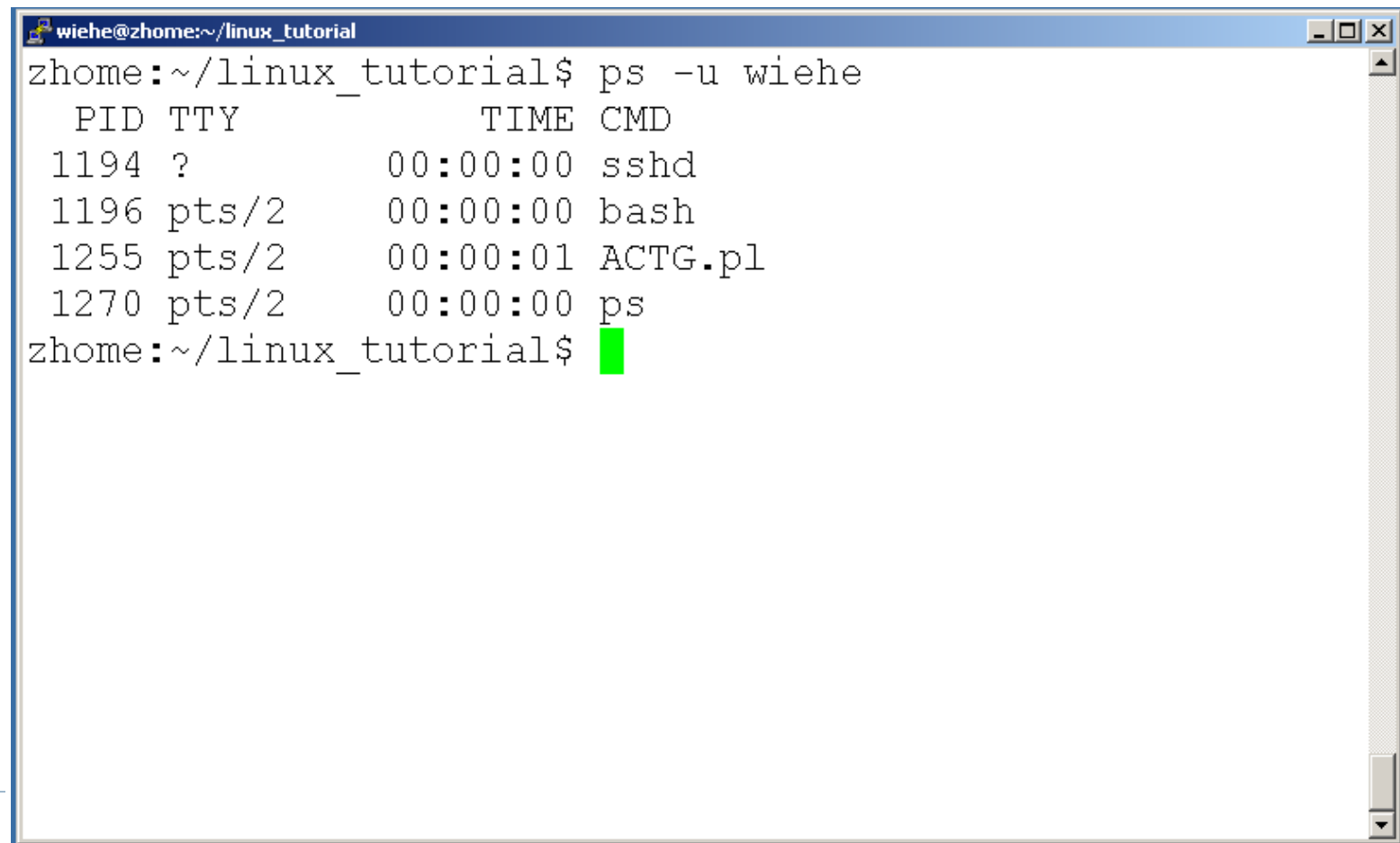
- ▶ To terminate a process use “kill”



```
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$ █
```

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$
```

The image shows a terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The terminal content displays the command 'ps -u wiehe' and its output, which is a table of processes. The table has four columns: PID, TTY, TIME, and CMD. The processes listed are sshd (PID 1194), bash (PID 1196), ACTG.pl (PID 1255), and ps (PID 1270). The terminal prompt 'zhome:~/linux_tutorial\$' is shown at the end of the output, followed by a green cursor block.

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