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

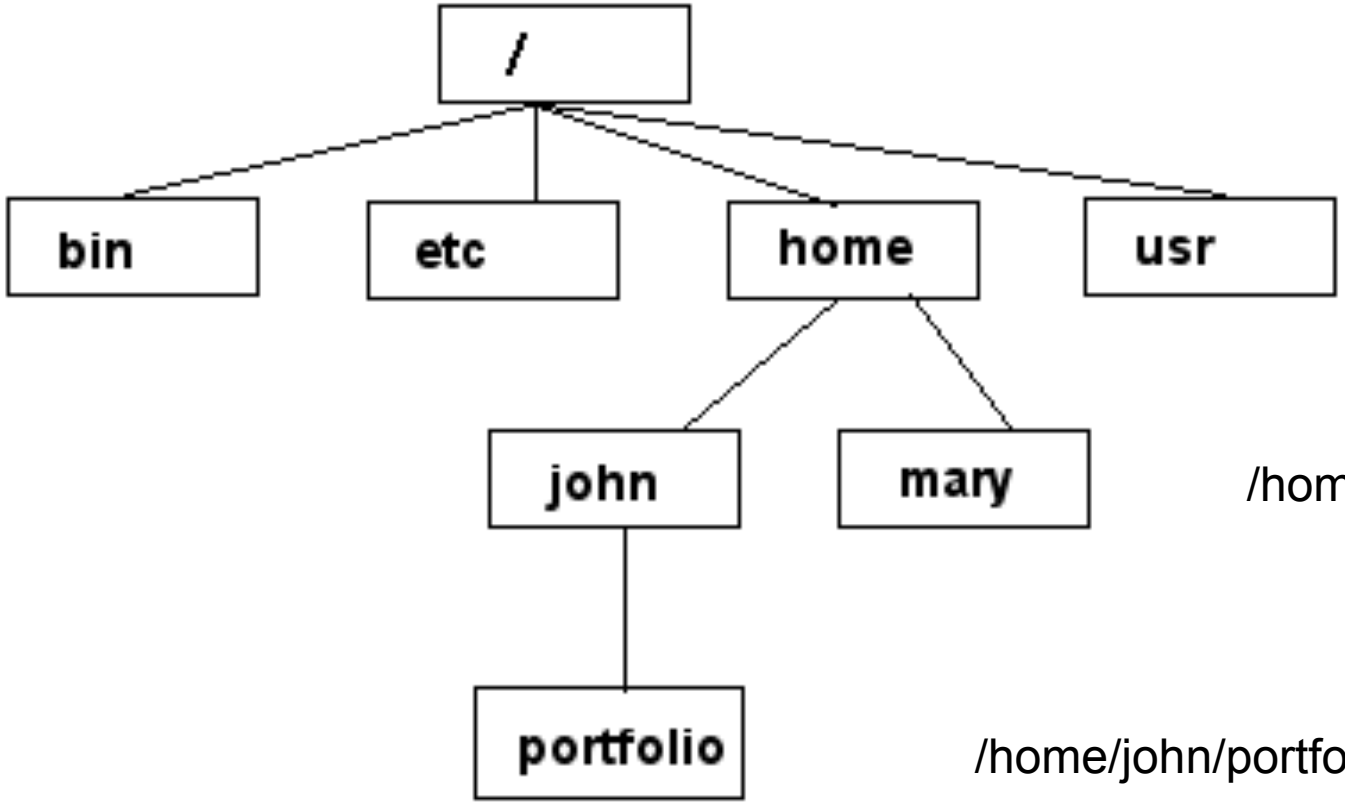
Introduction: Why Unix/Linux?

- 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!**



/home/mary/

/home/john/portfolio/

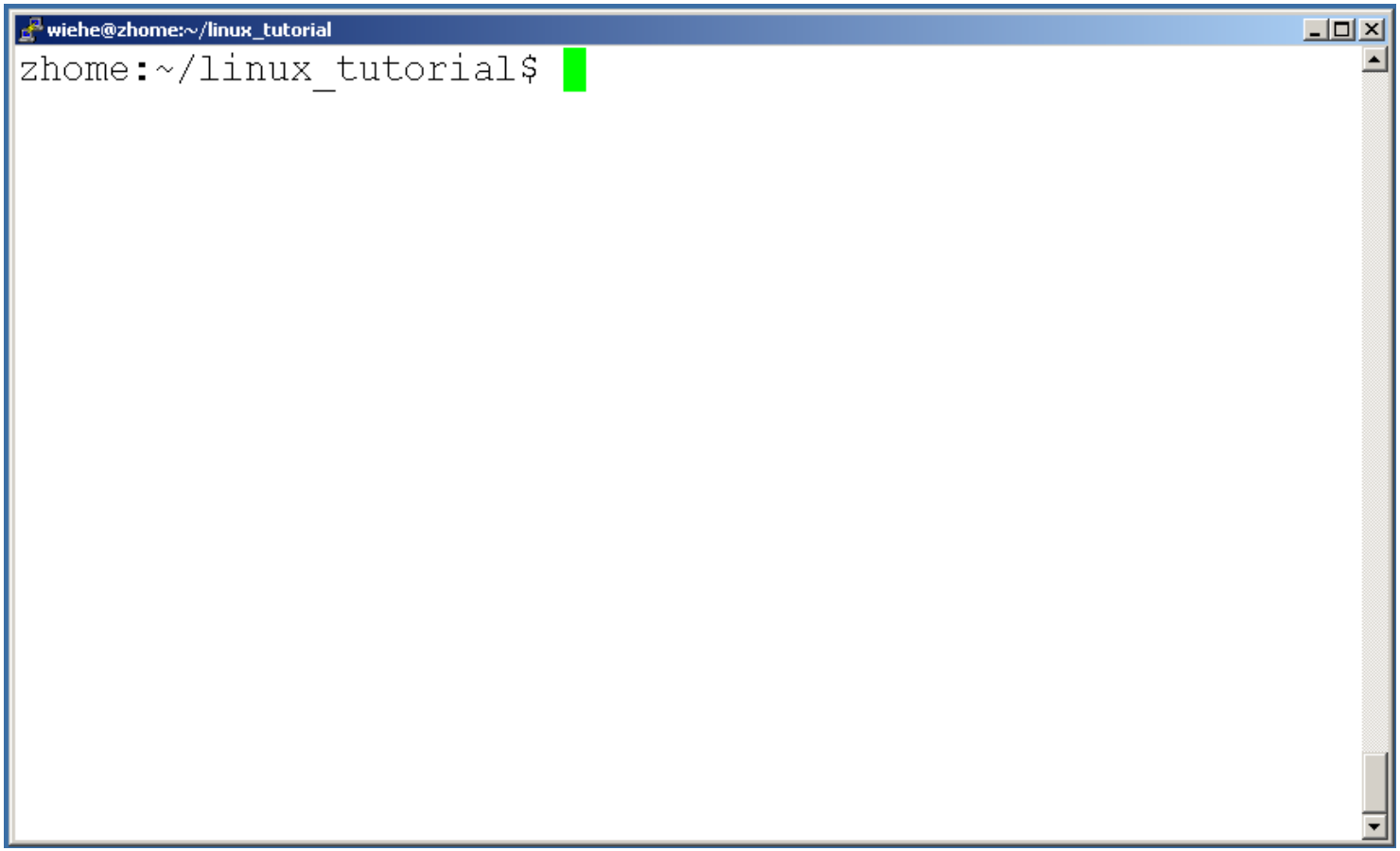
The Path

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!**

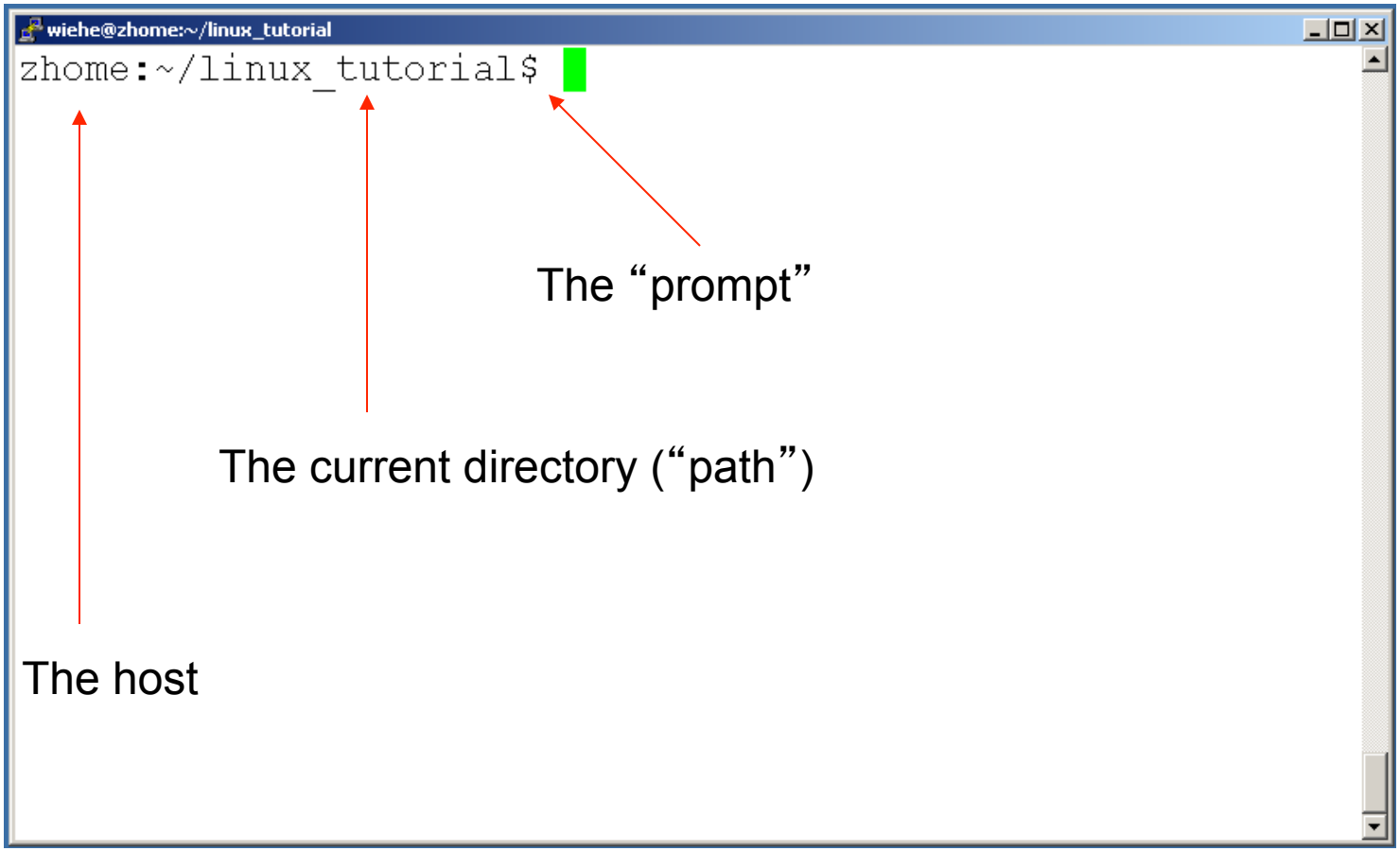
Connecting to a Unix/Linux system

- Open up a terminal:

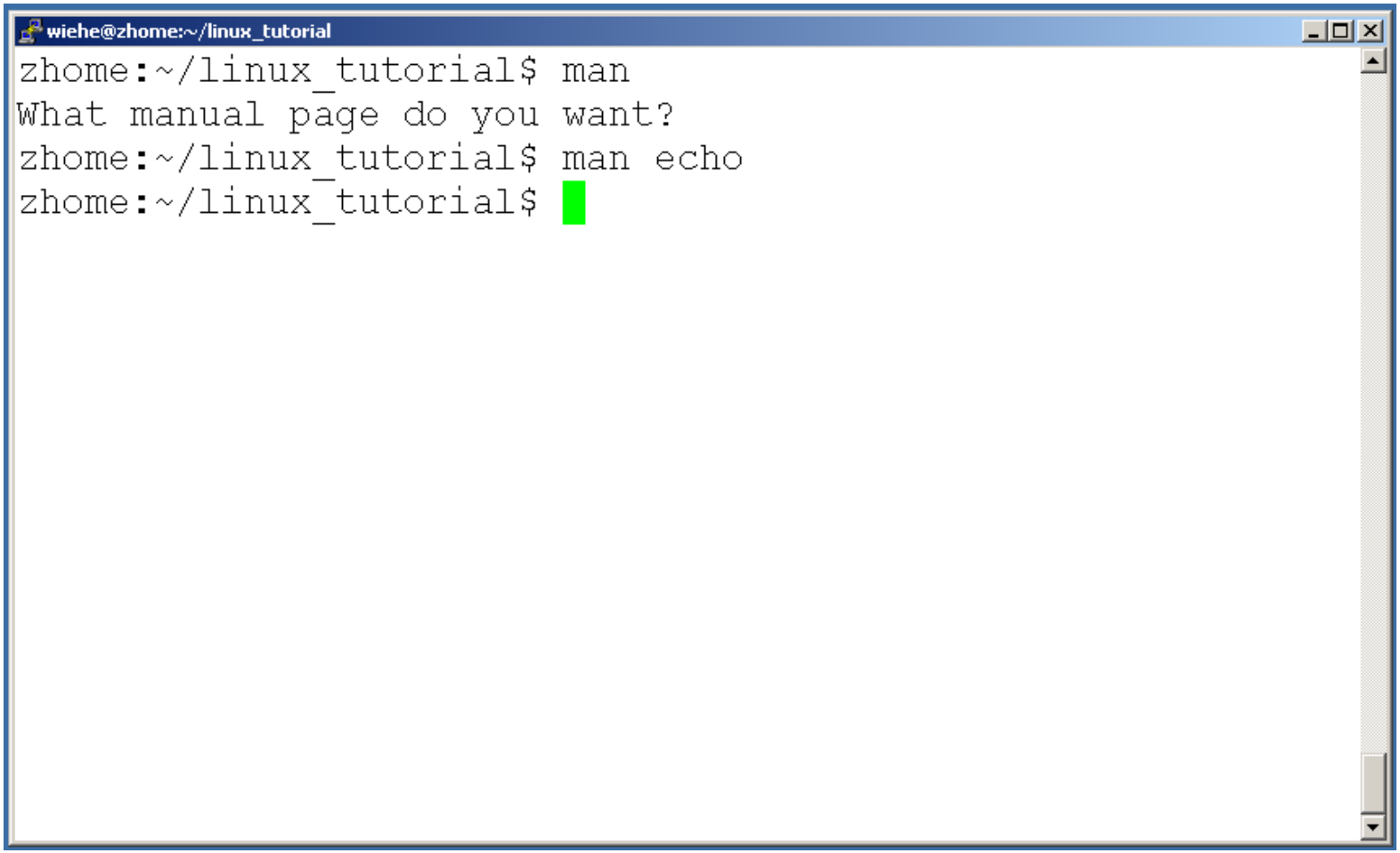


Connecting to a Unix/Linux system

- Open up a terminal:



Help!

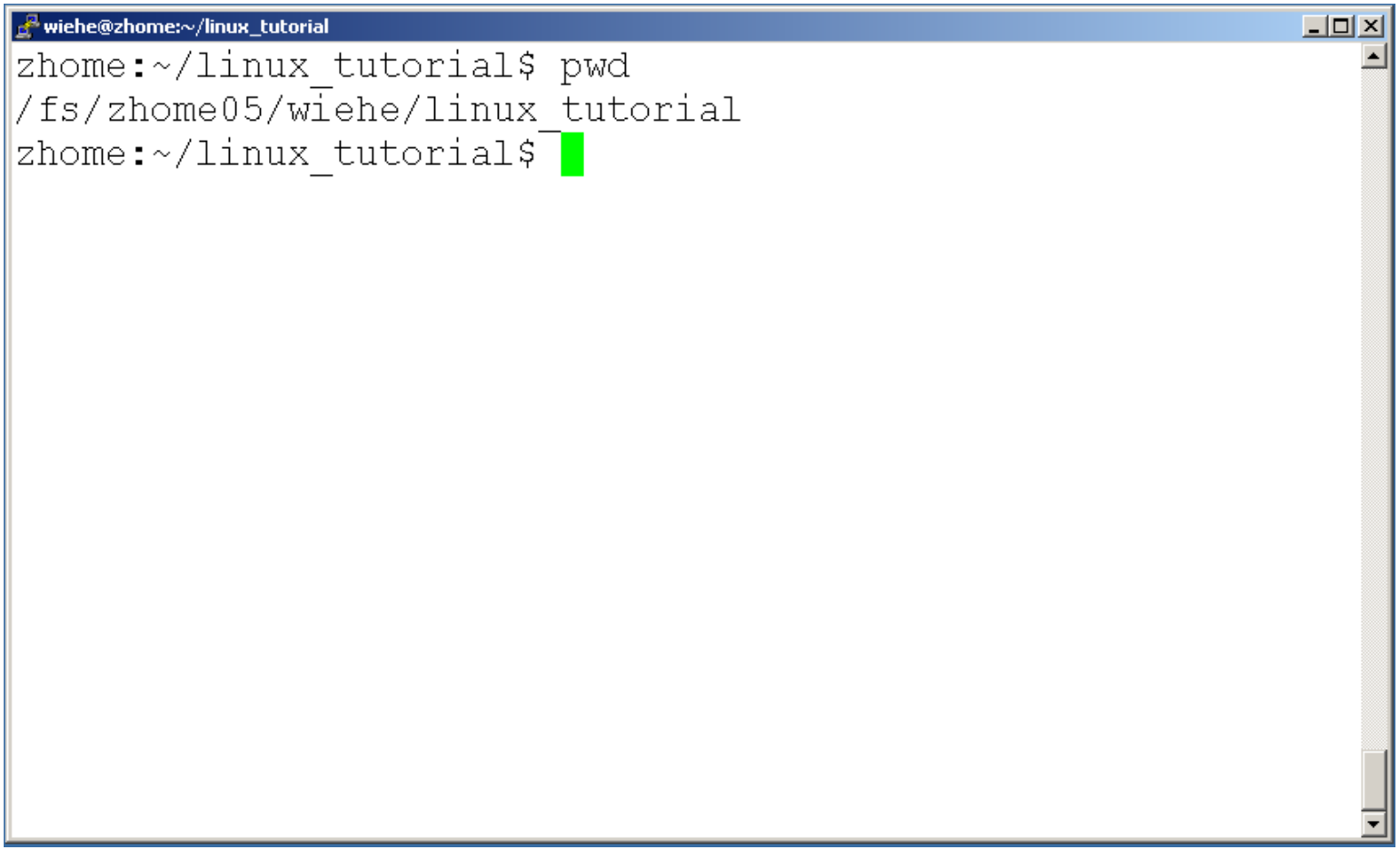

wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ man
What manual page do you want?
zhome:~/linux_tutorial$ man echo
zhome:~/linux_tutorial$

Help!

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

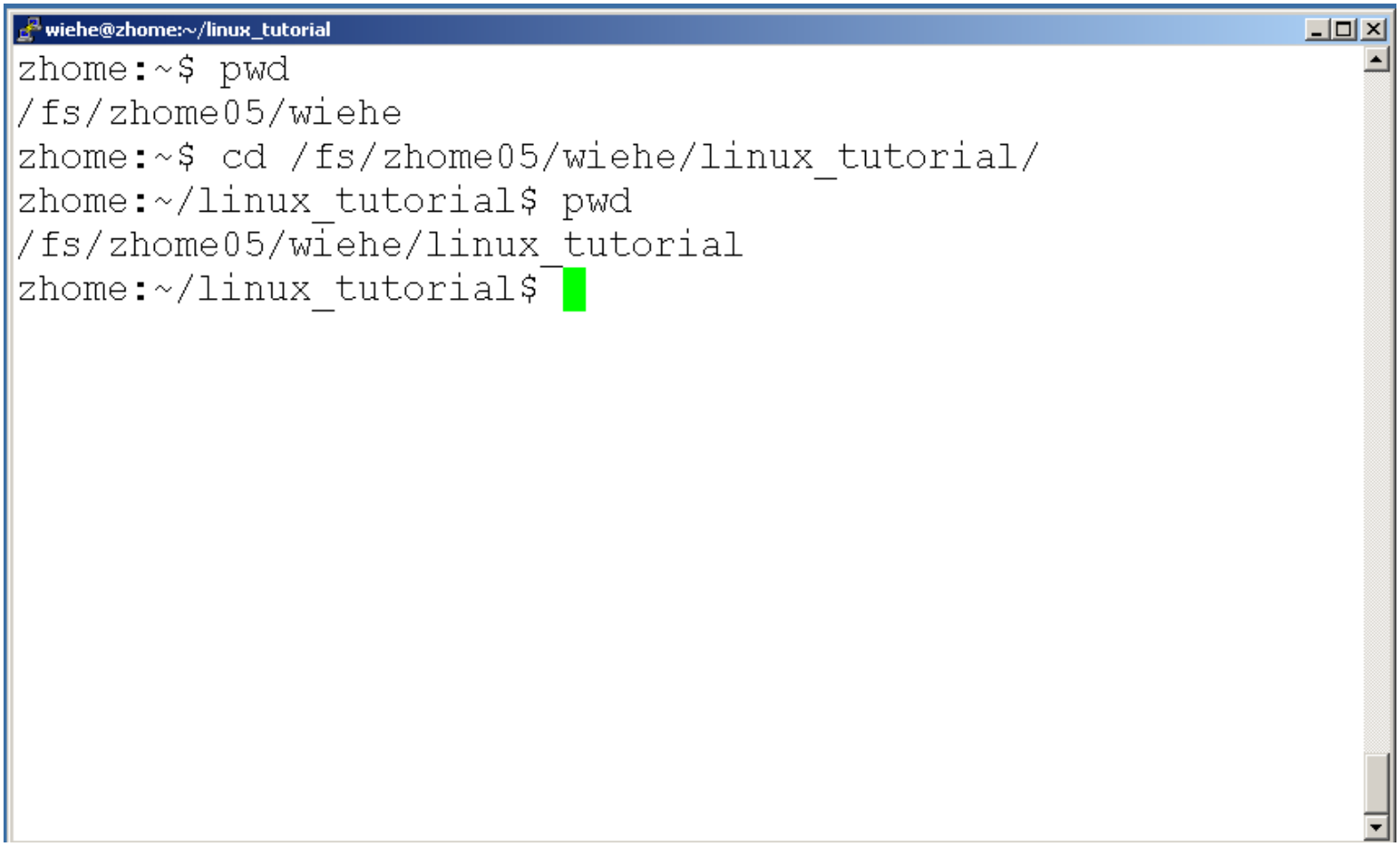
Command: pwd

- To find your current path use “pwd”



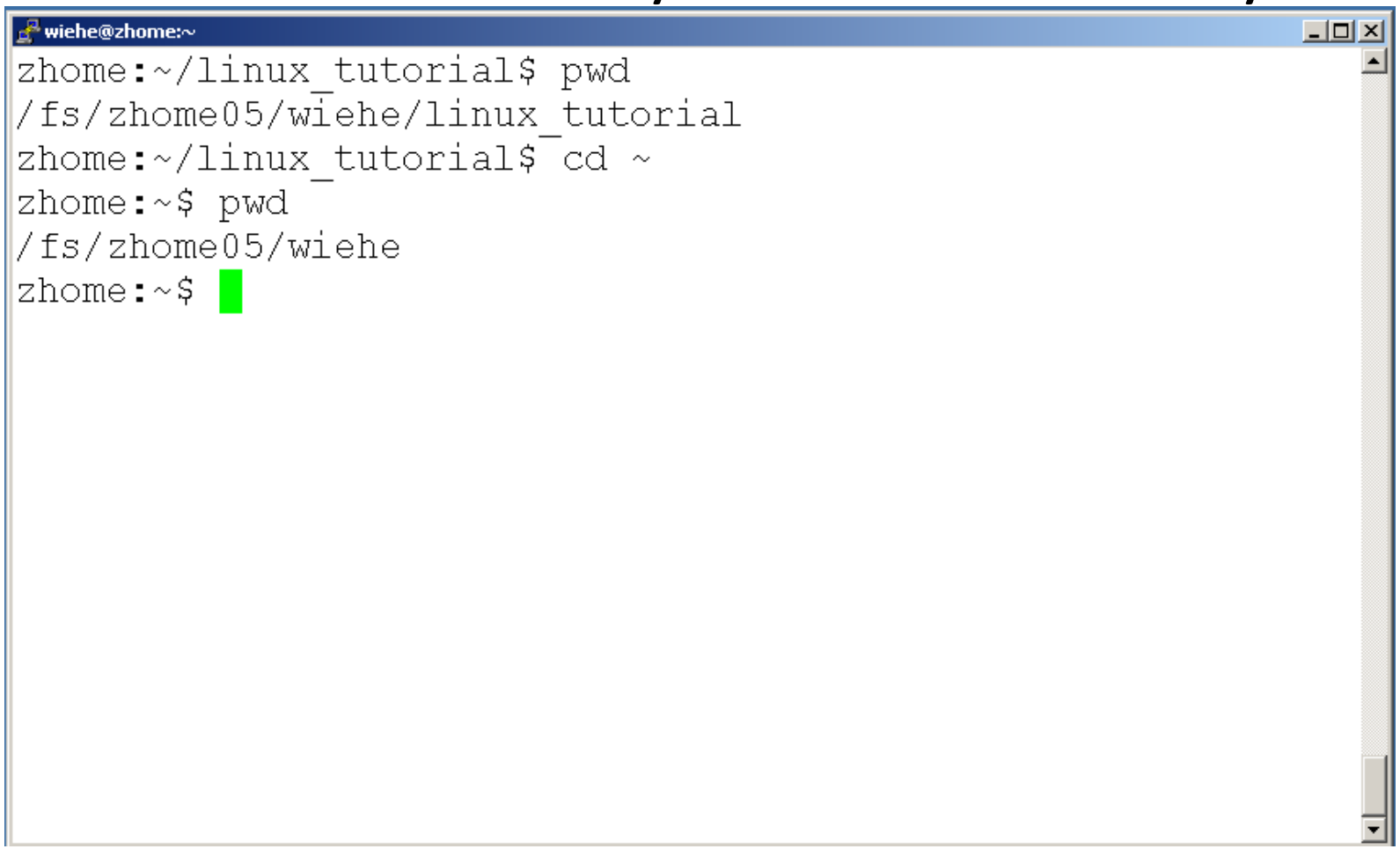
Command: cd

- To change to a specific directory use "cd"



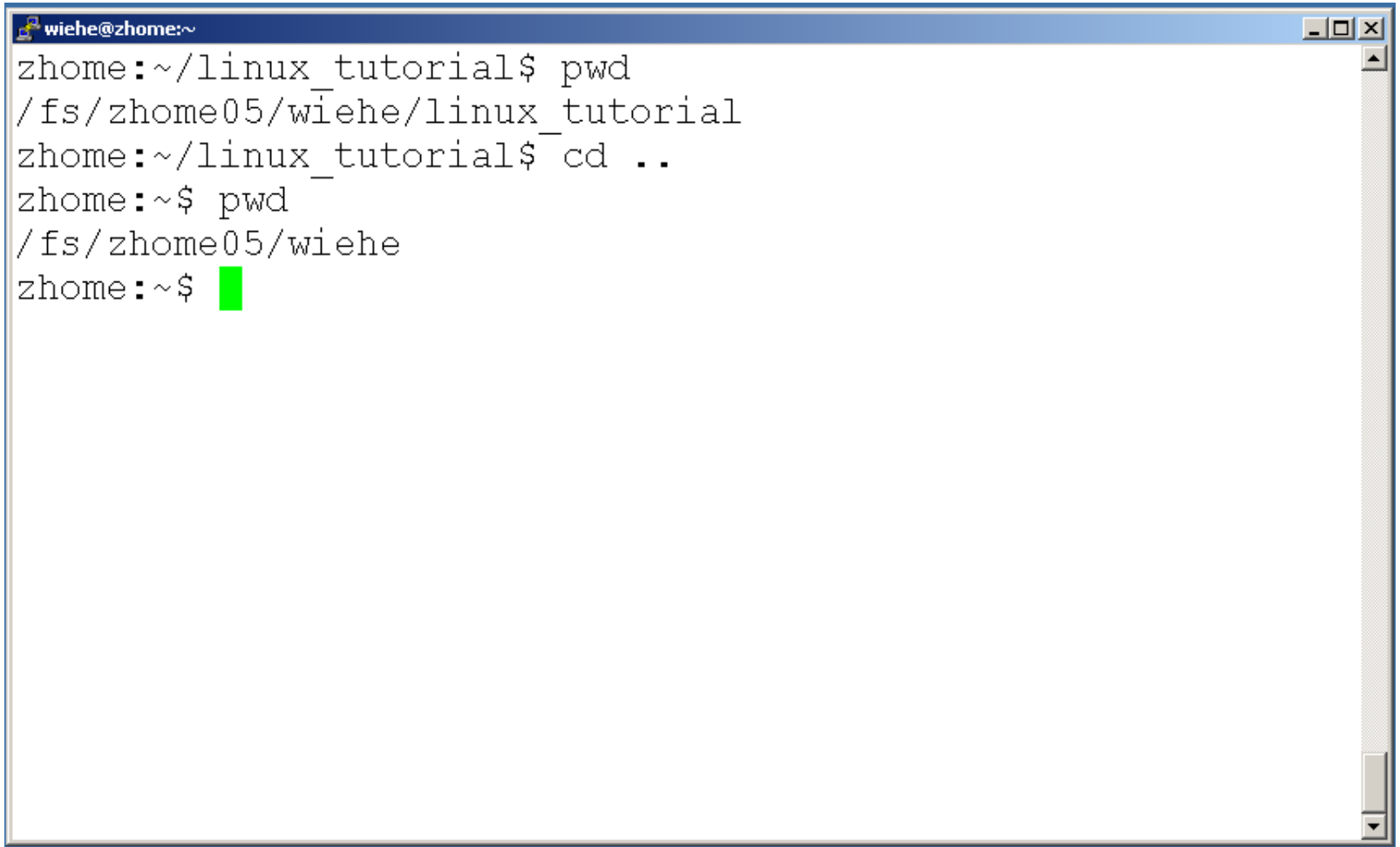
Command: cd

- "~" is the location of your home directory



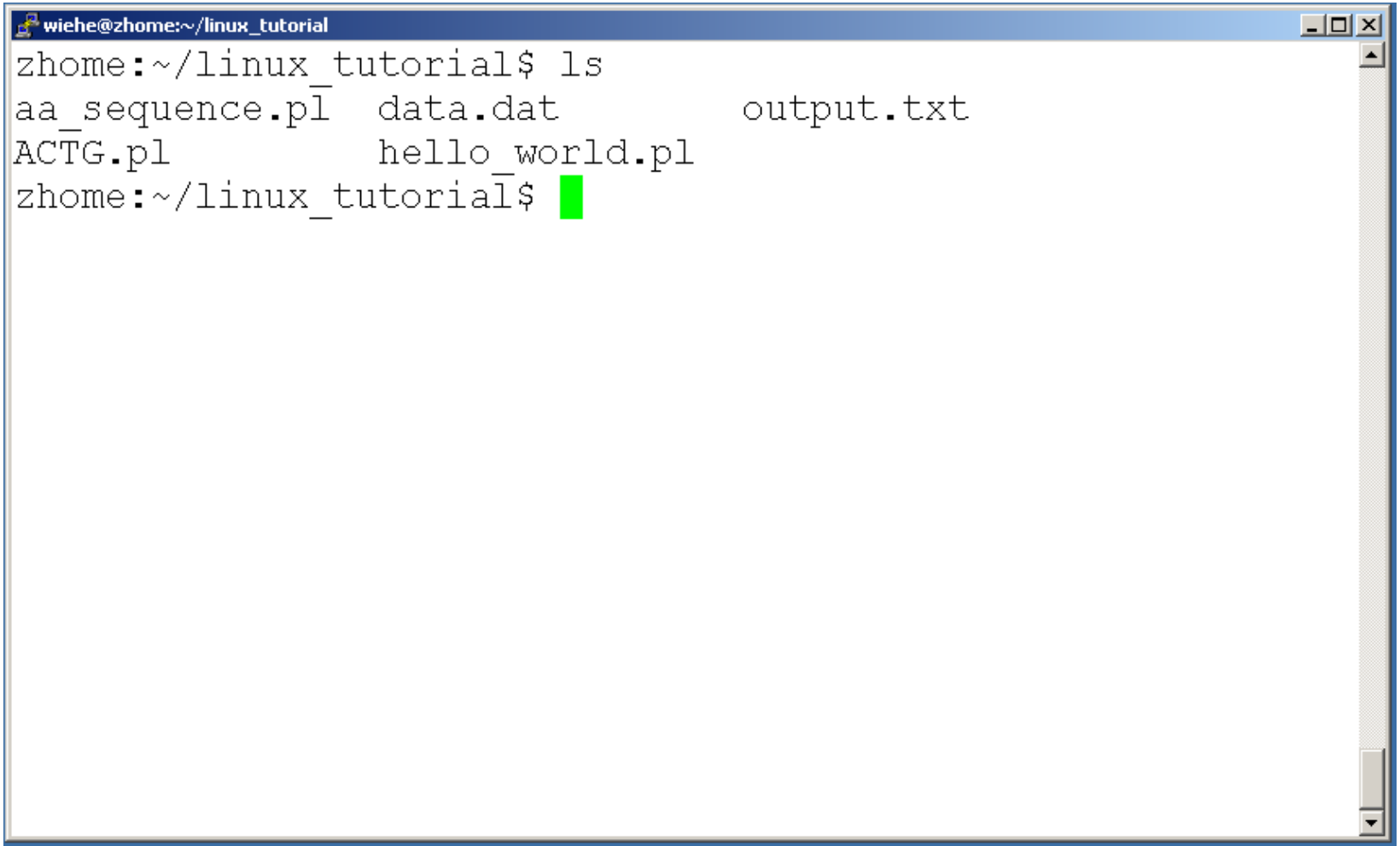
Command: cd

- "../" is the location of the directory below current one



Command: ls

- To list the files in the current directory use "ls"

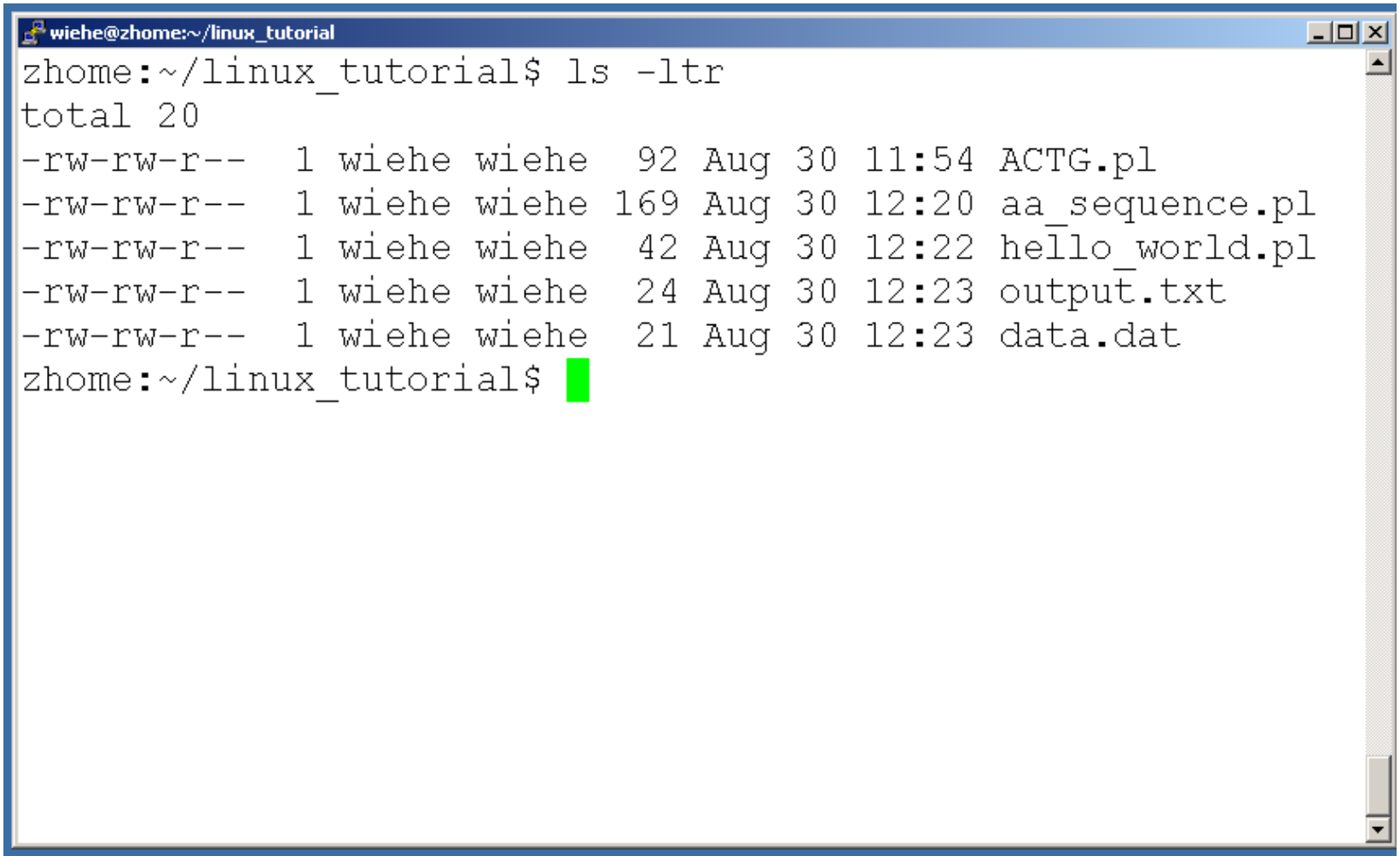


Command: ls

- 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”

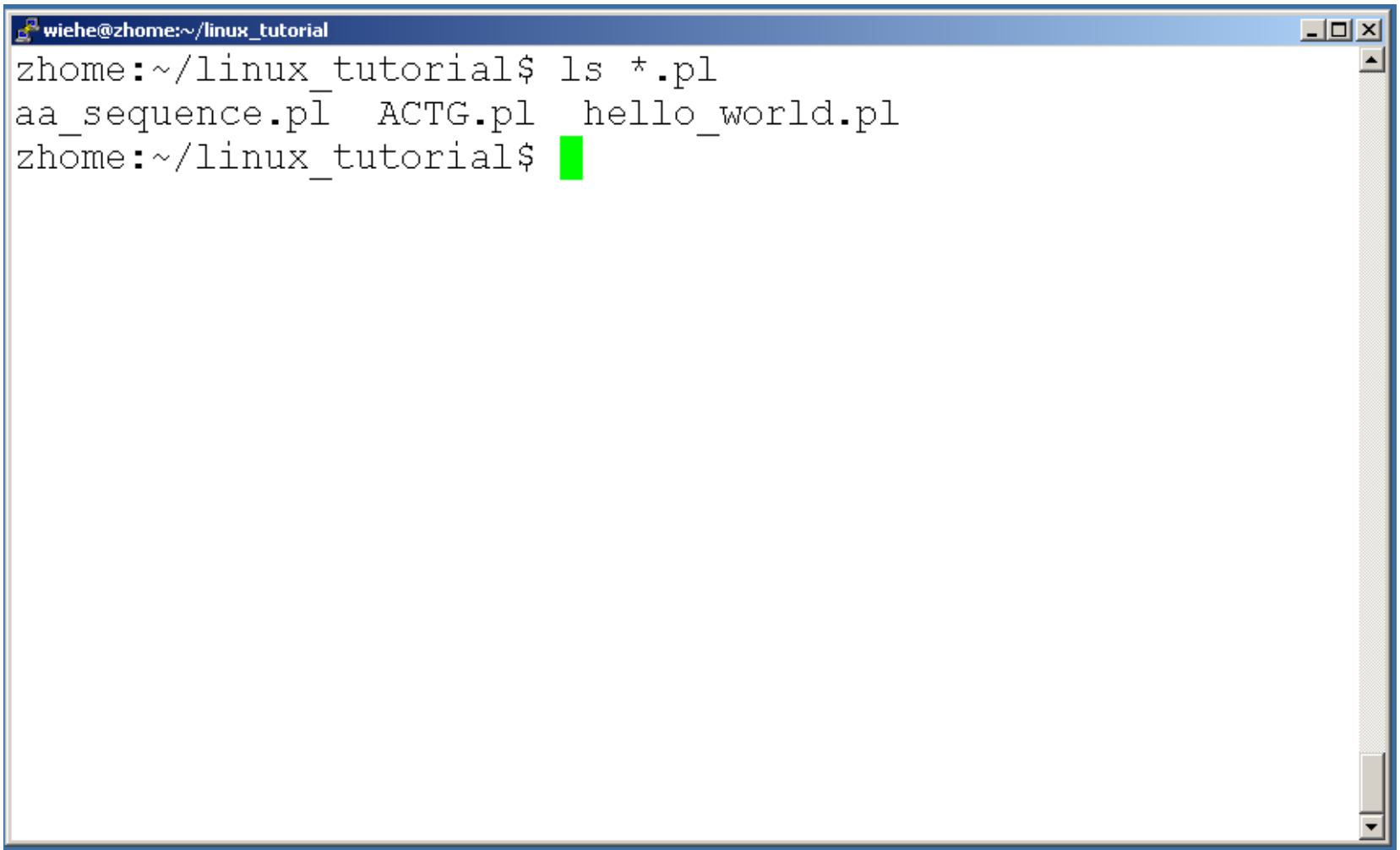
Command: ls -ltr

- List files by time in reverse order with long listing



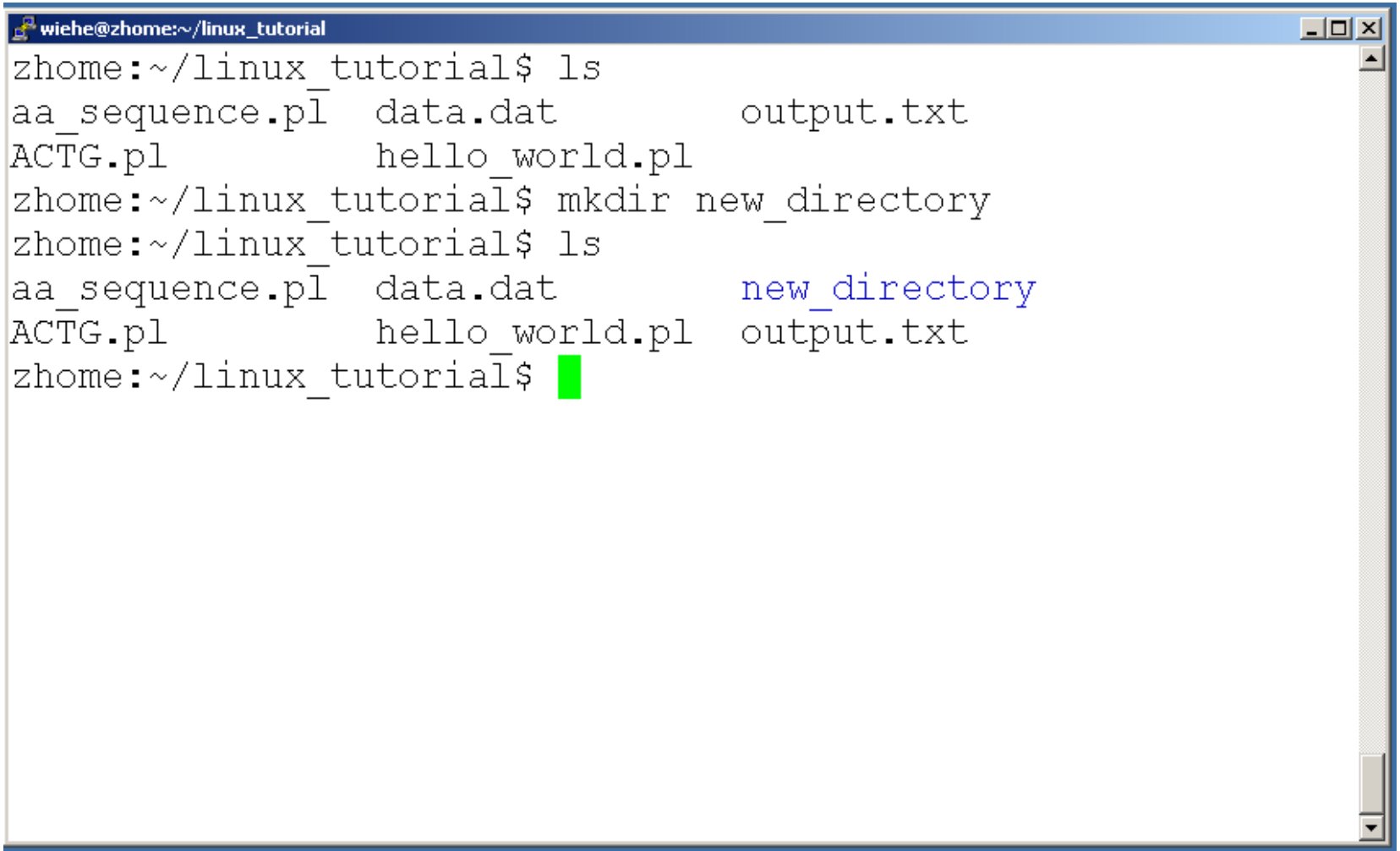
General Syntax: *

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



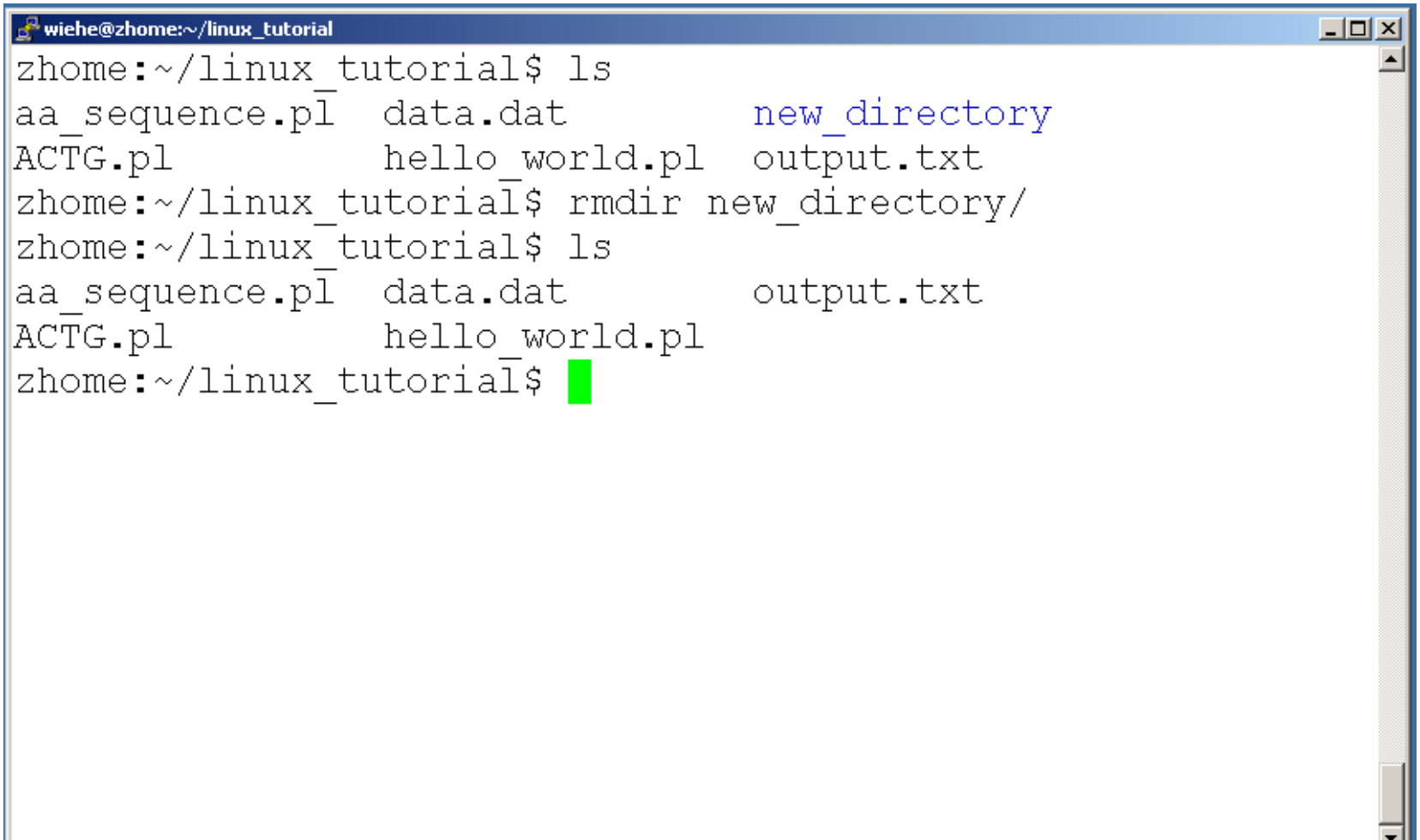
Command: mkdir

- To create a new directory use “mkdir”



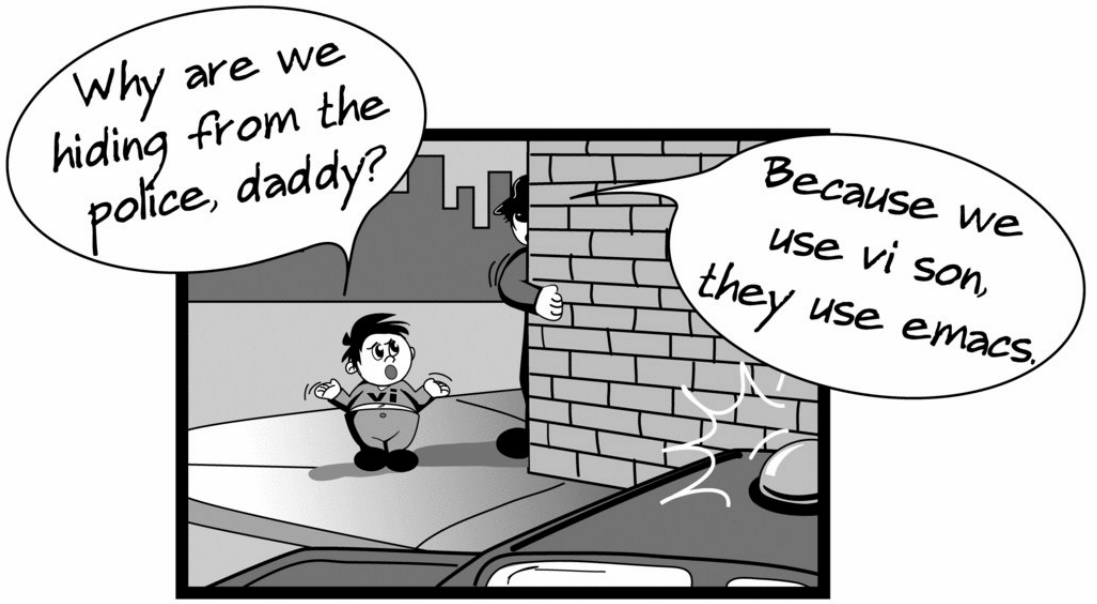
Command: rmdir

- To remove and empty directory use “rmdir”



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

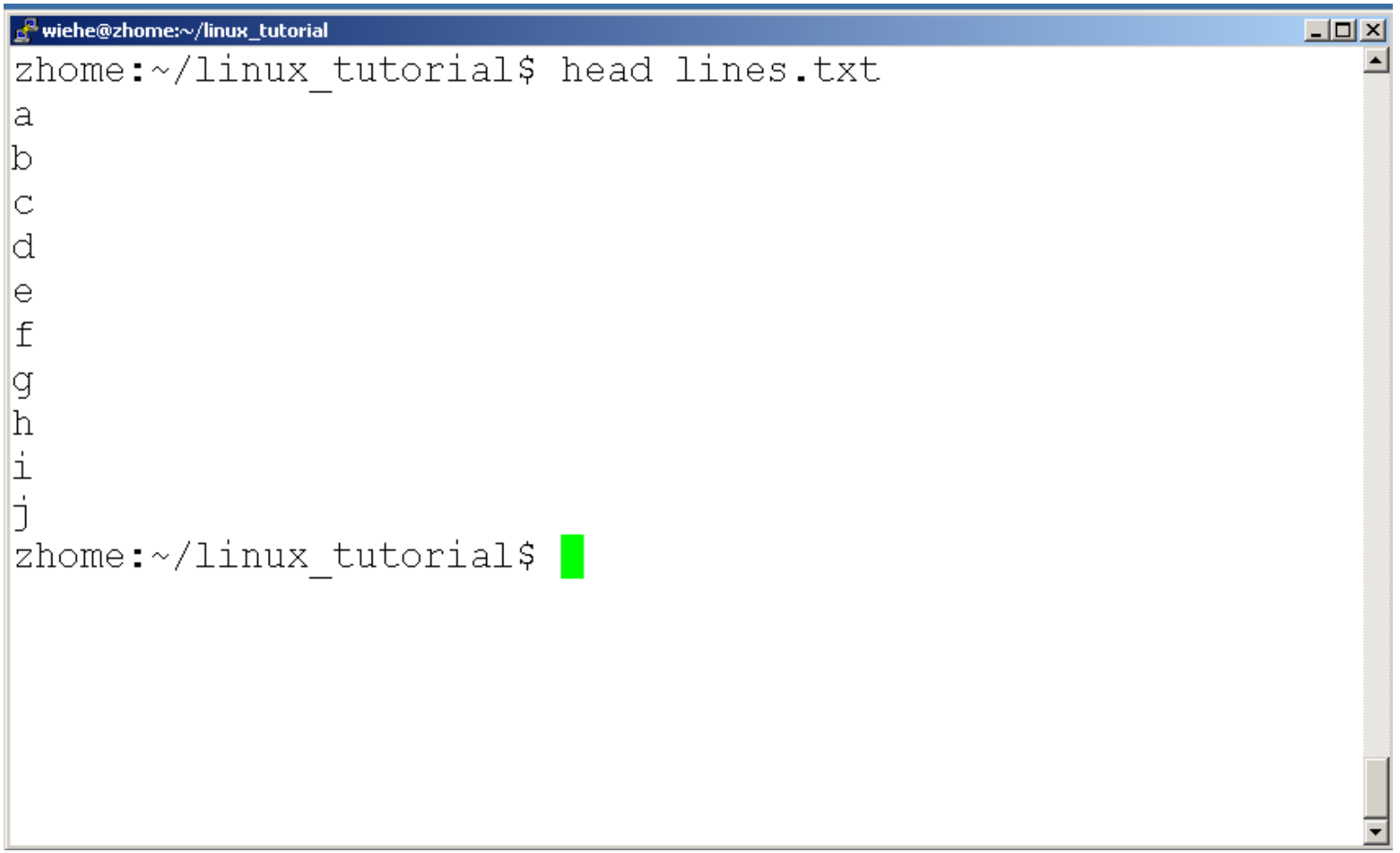
- “less” displays a file, allowing forward/ backward movement within it
  - return scrolls forward one line, space one page
  - y scrolls back one line, b one page
- use “/” to search for a string
- Press q to quit

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

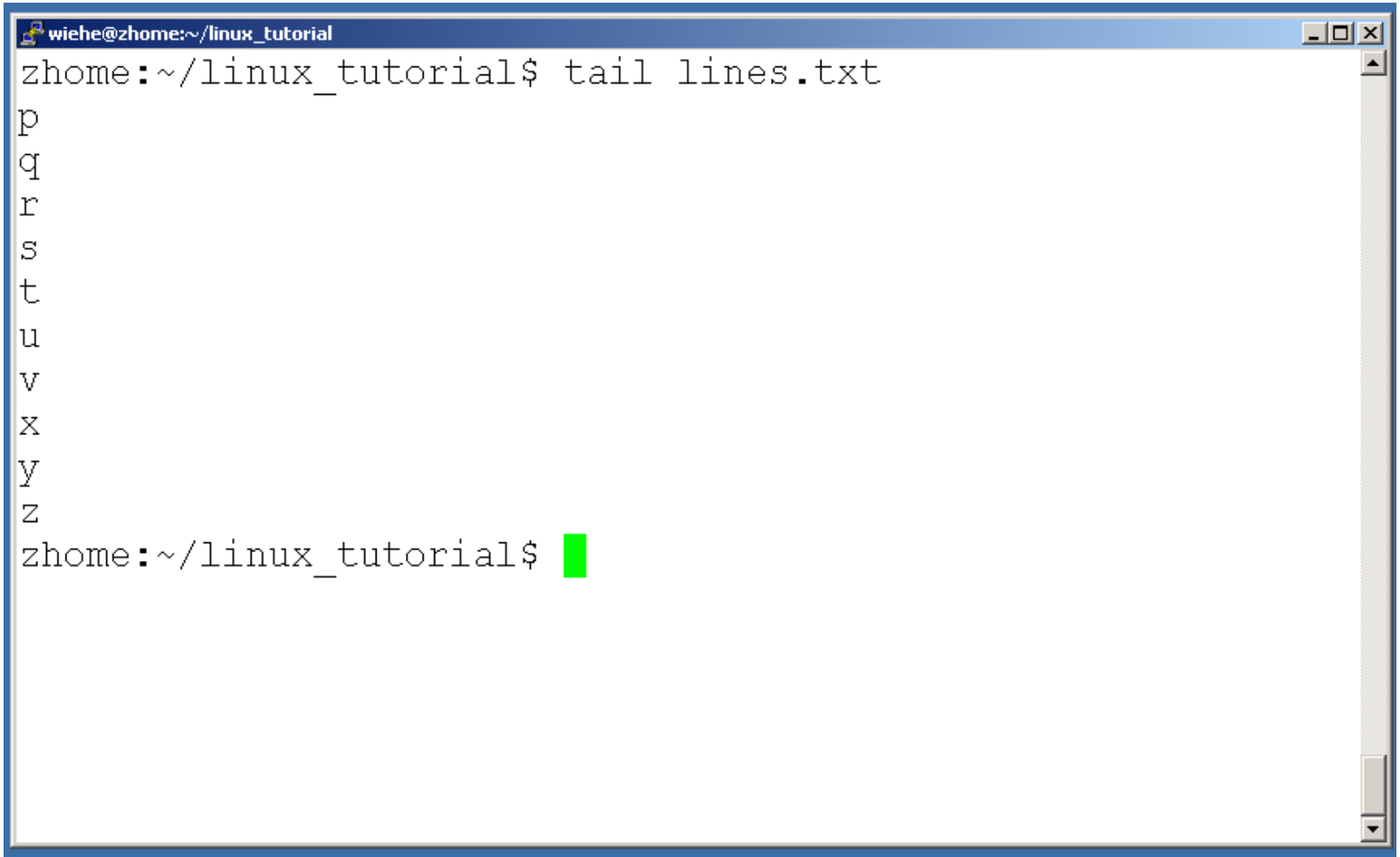
Command: head

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



Command: tail

- Same as head, but shows the last lines

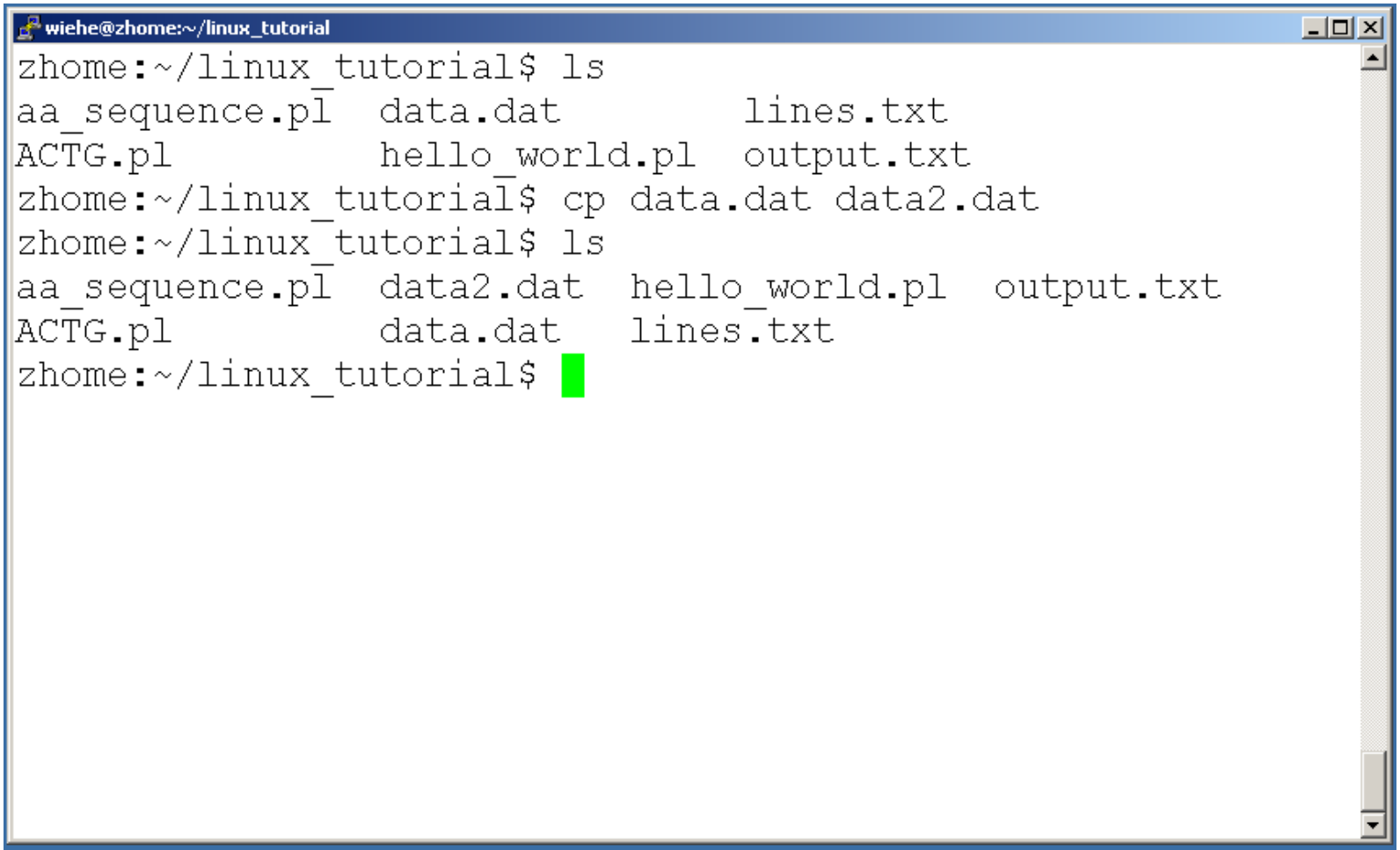


File Commands

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

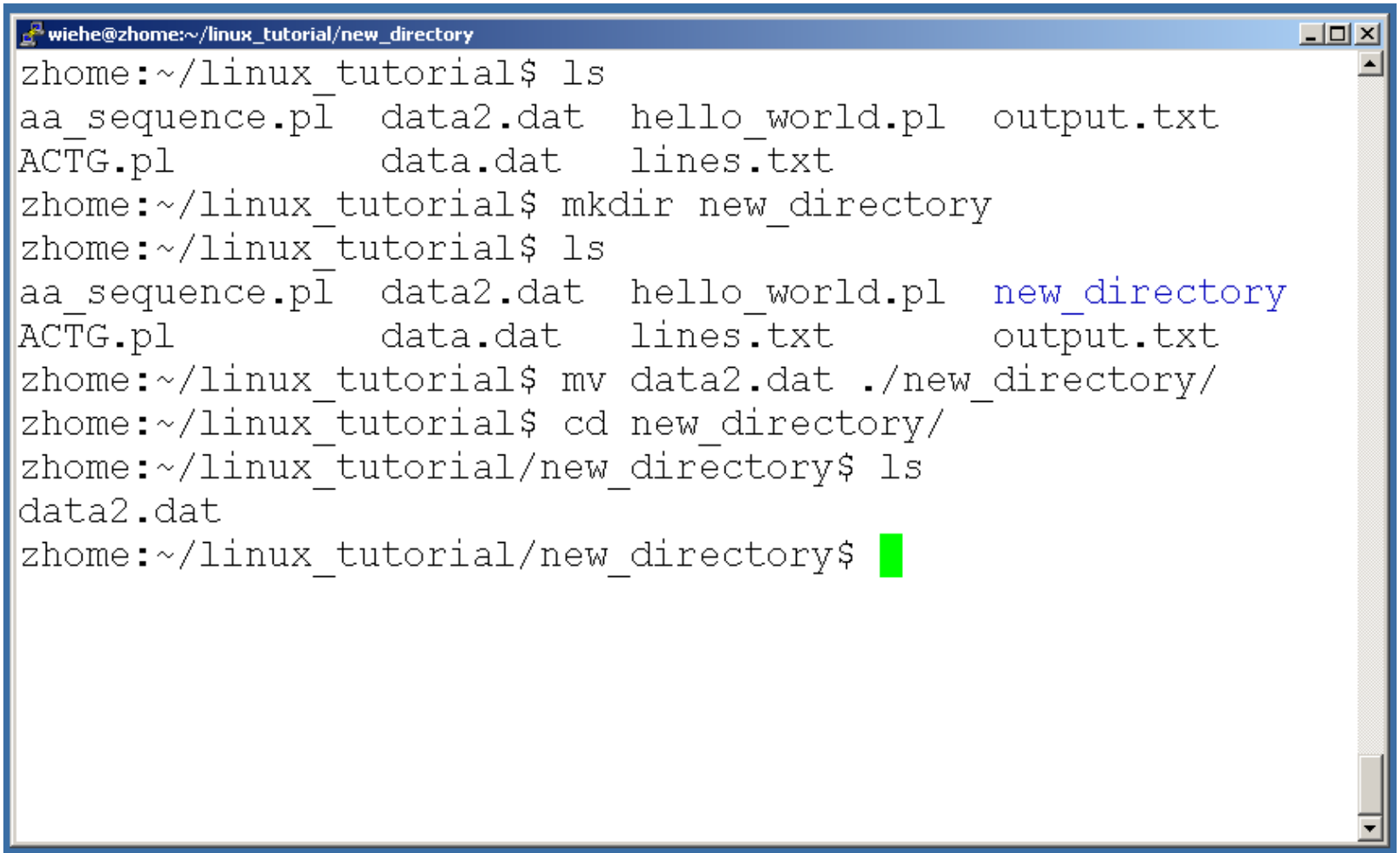
Command: cp

- To copy a file use "cp"



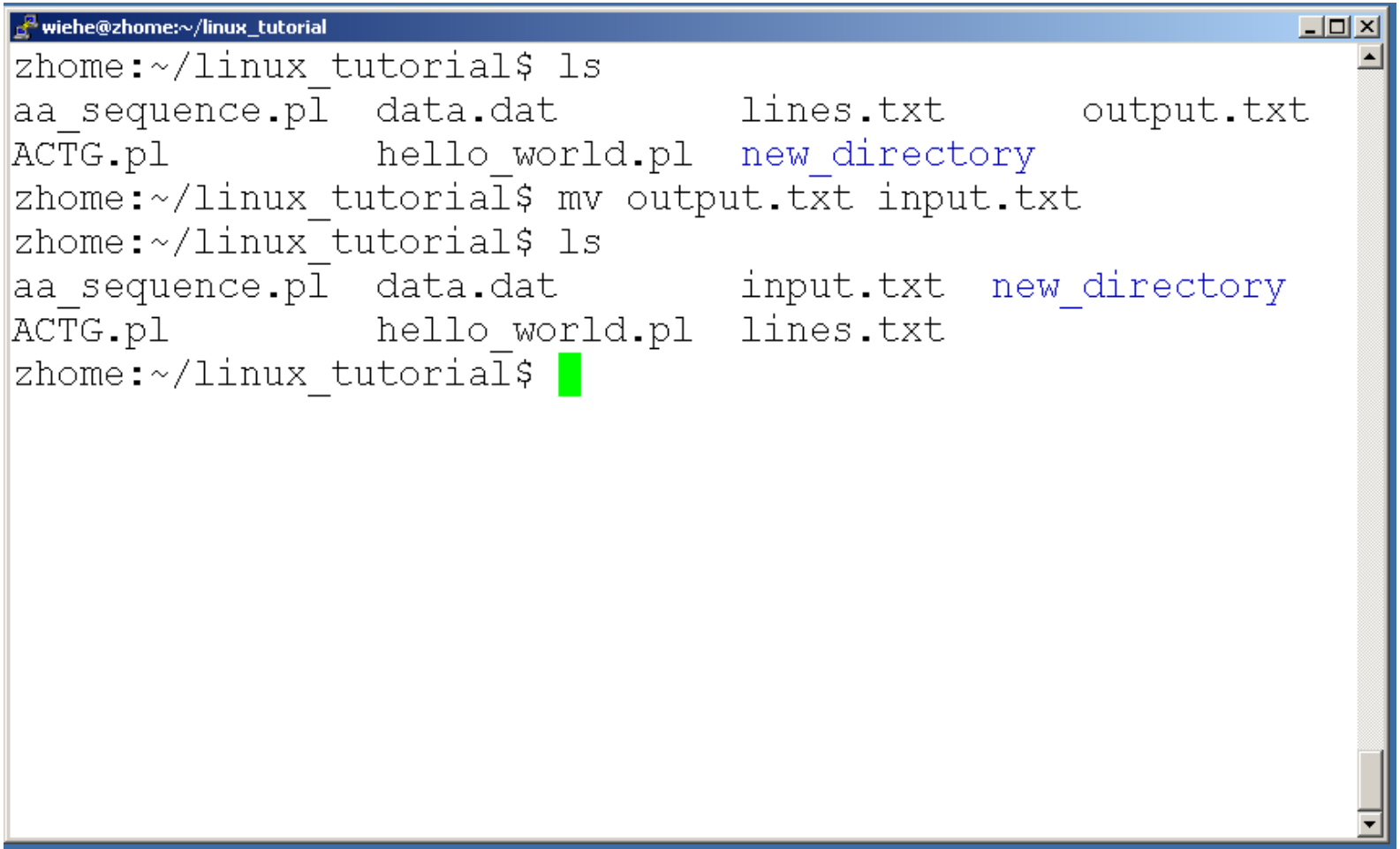
Command: mv

- To move a file to a different location use "mv"



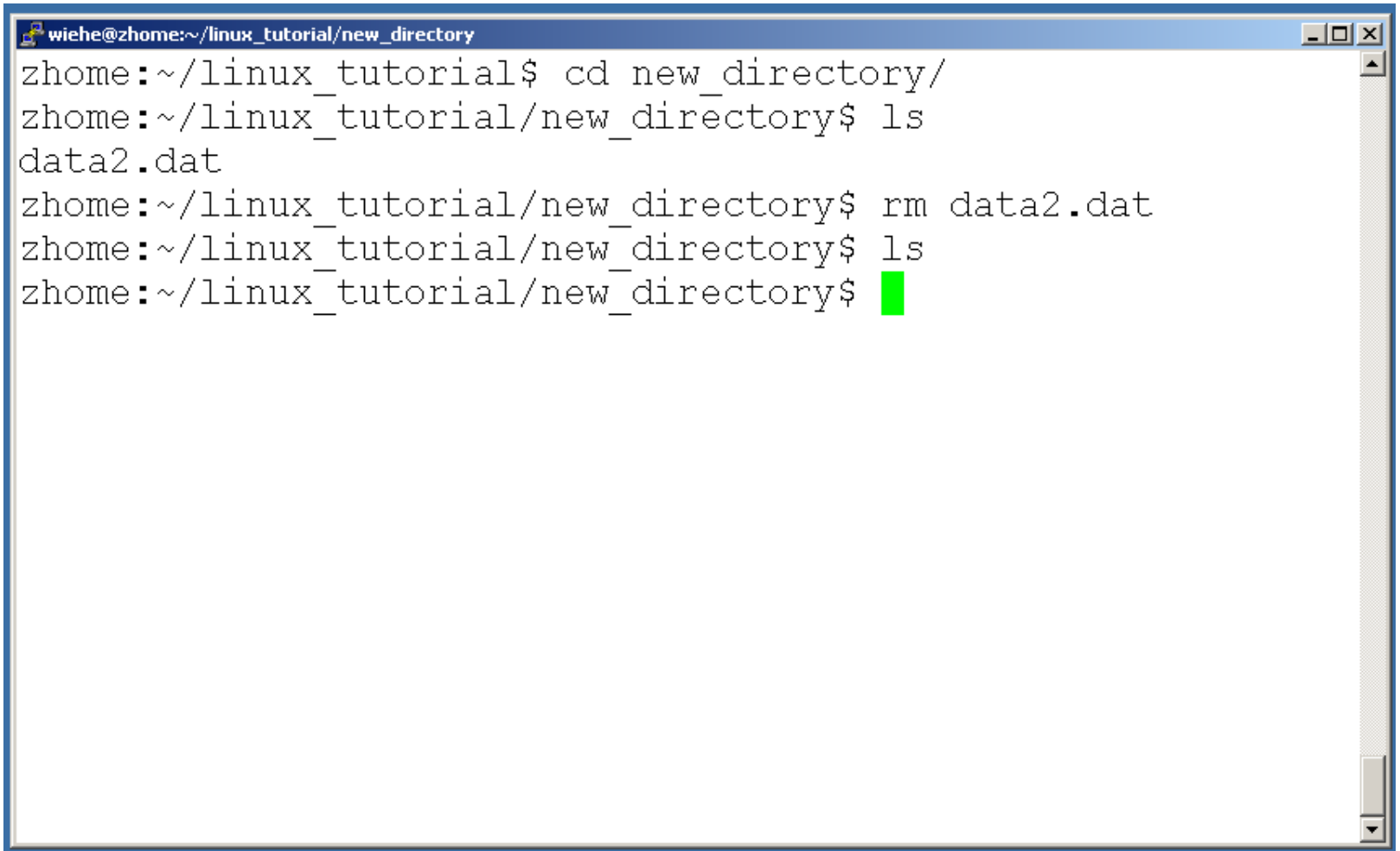
Command: mv

- mv can also be used to rename a file



Command: rm

- To remove a file use “rm”



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

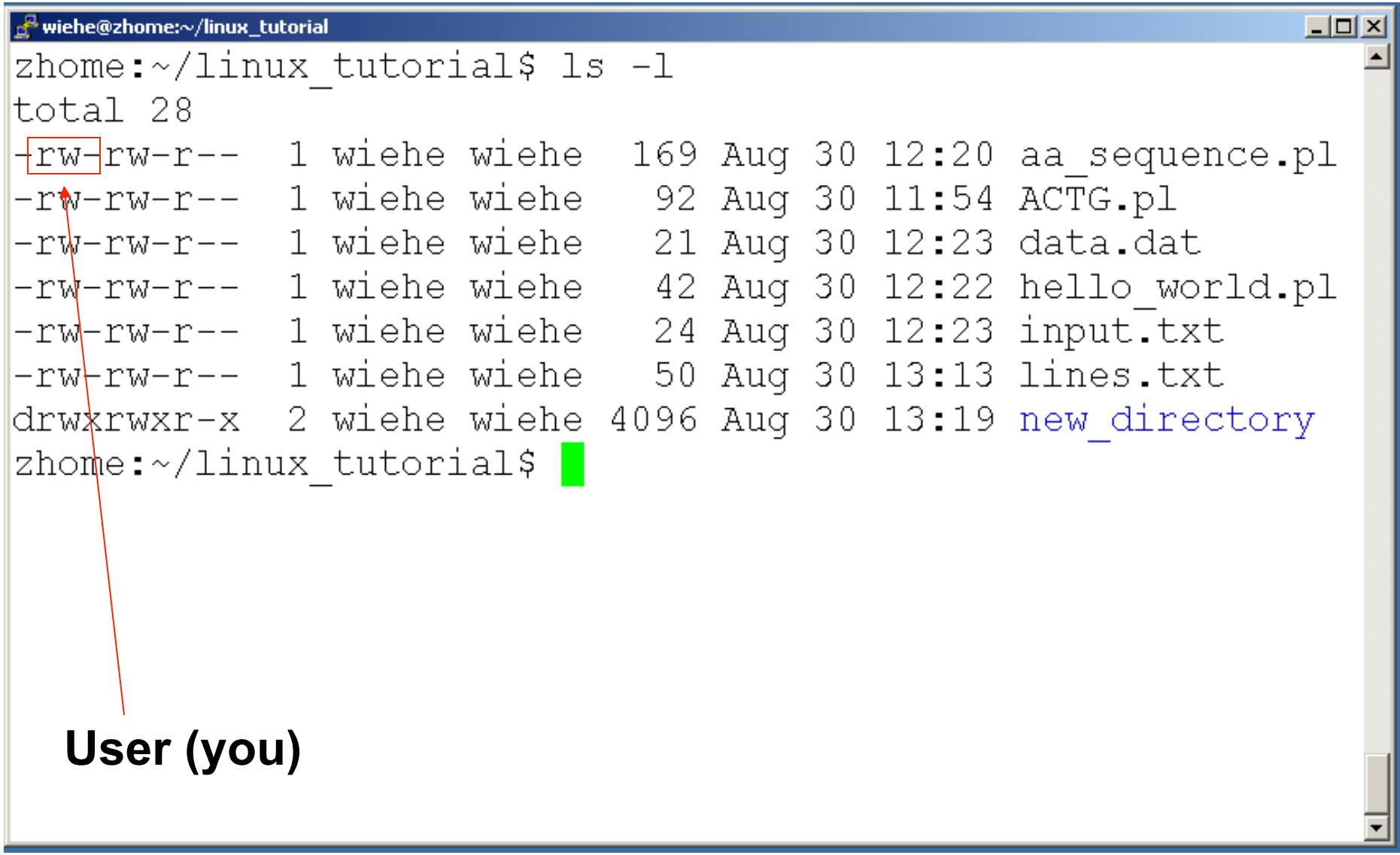
File permissions

- 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

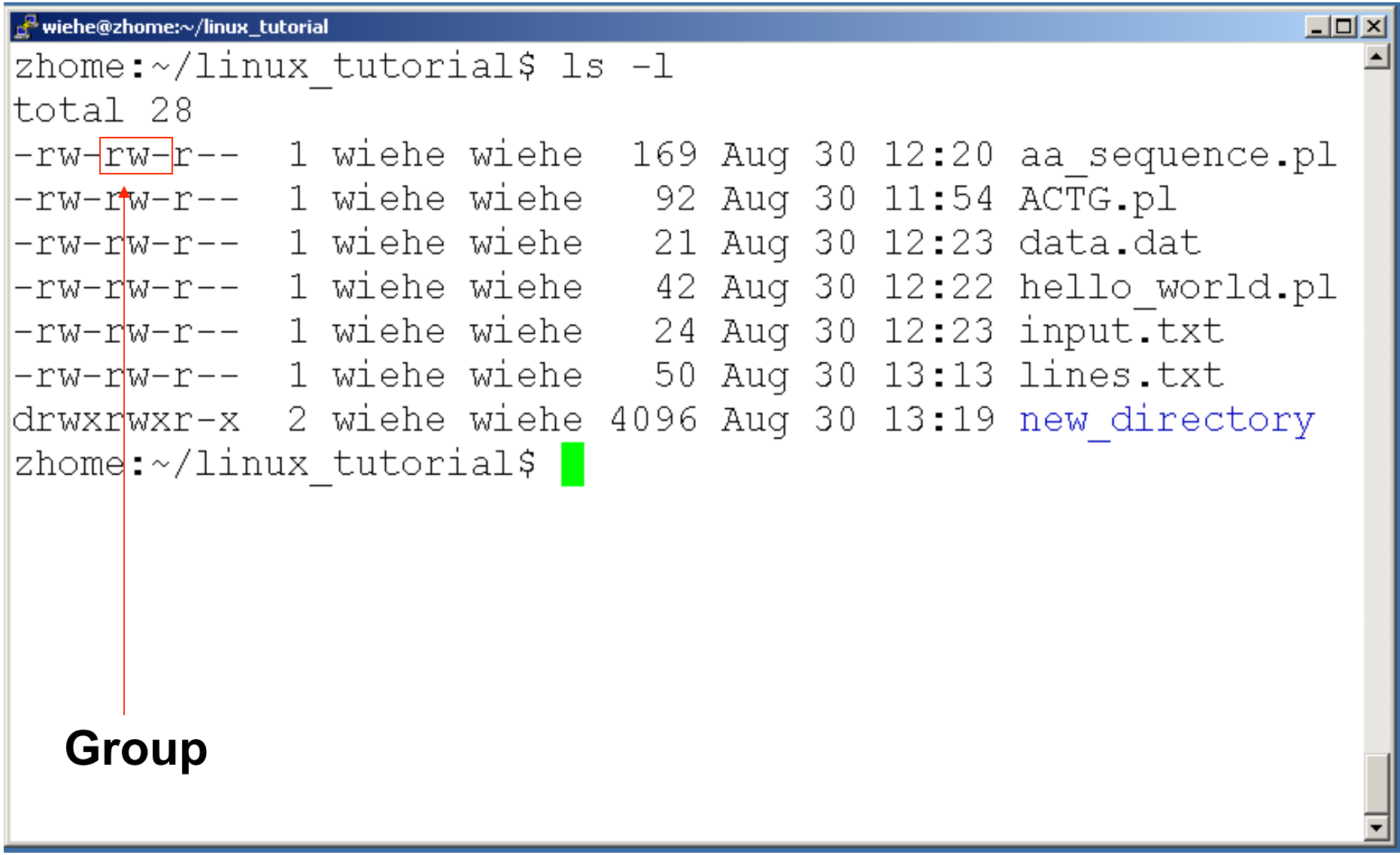
Permission levels

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

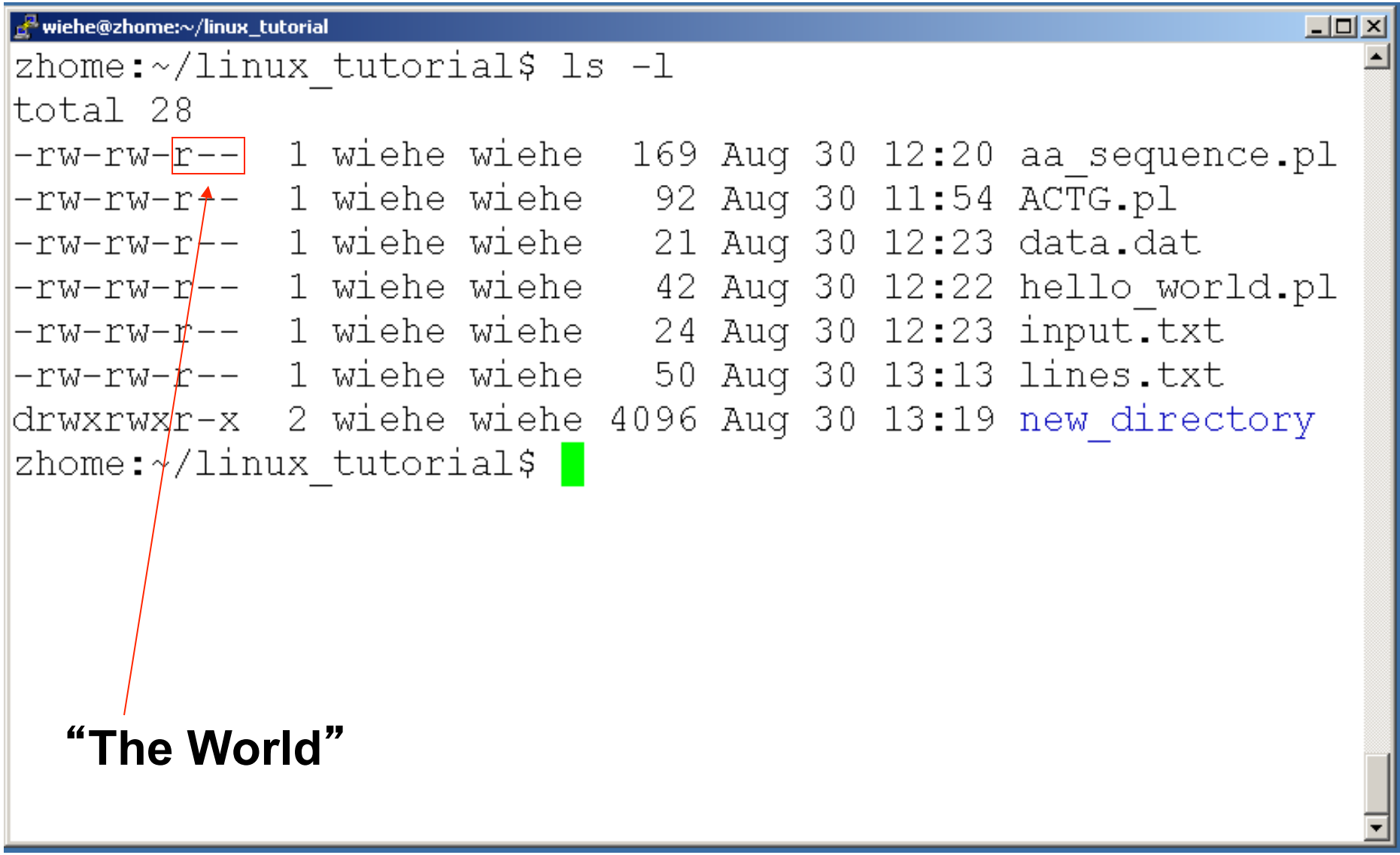
File Permissions



File Permissions

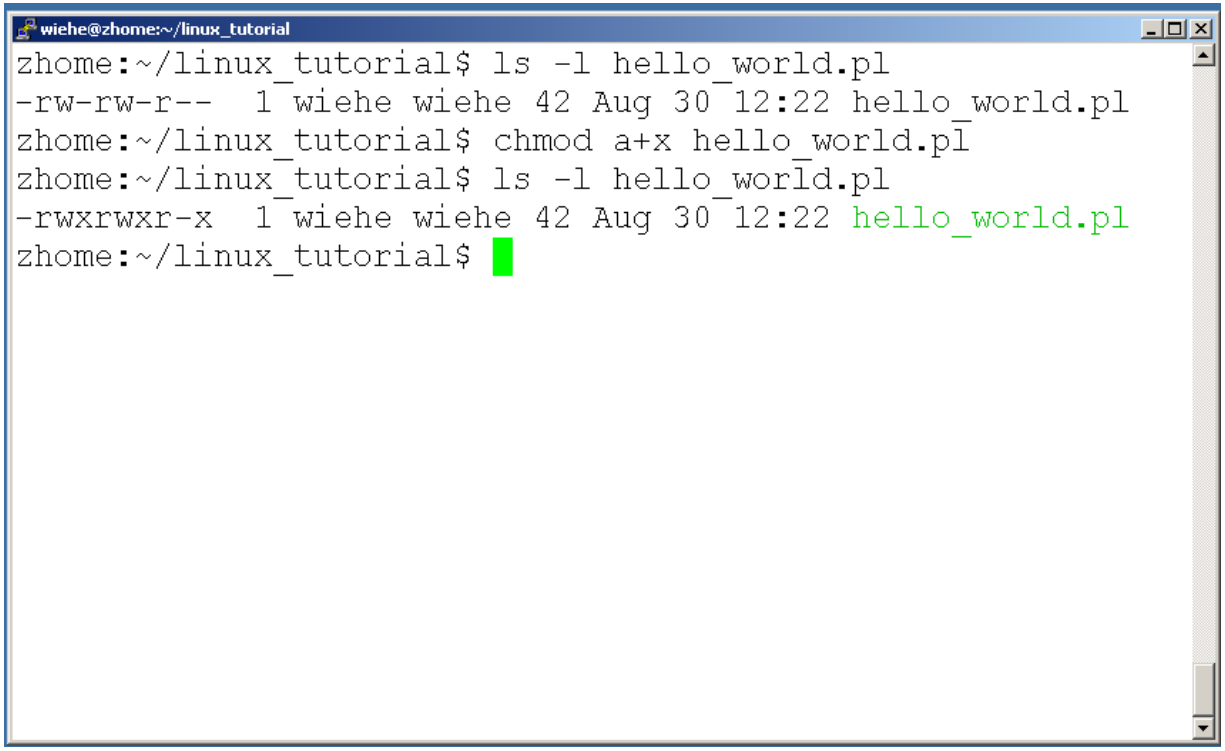


File Permissions



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:



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”