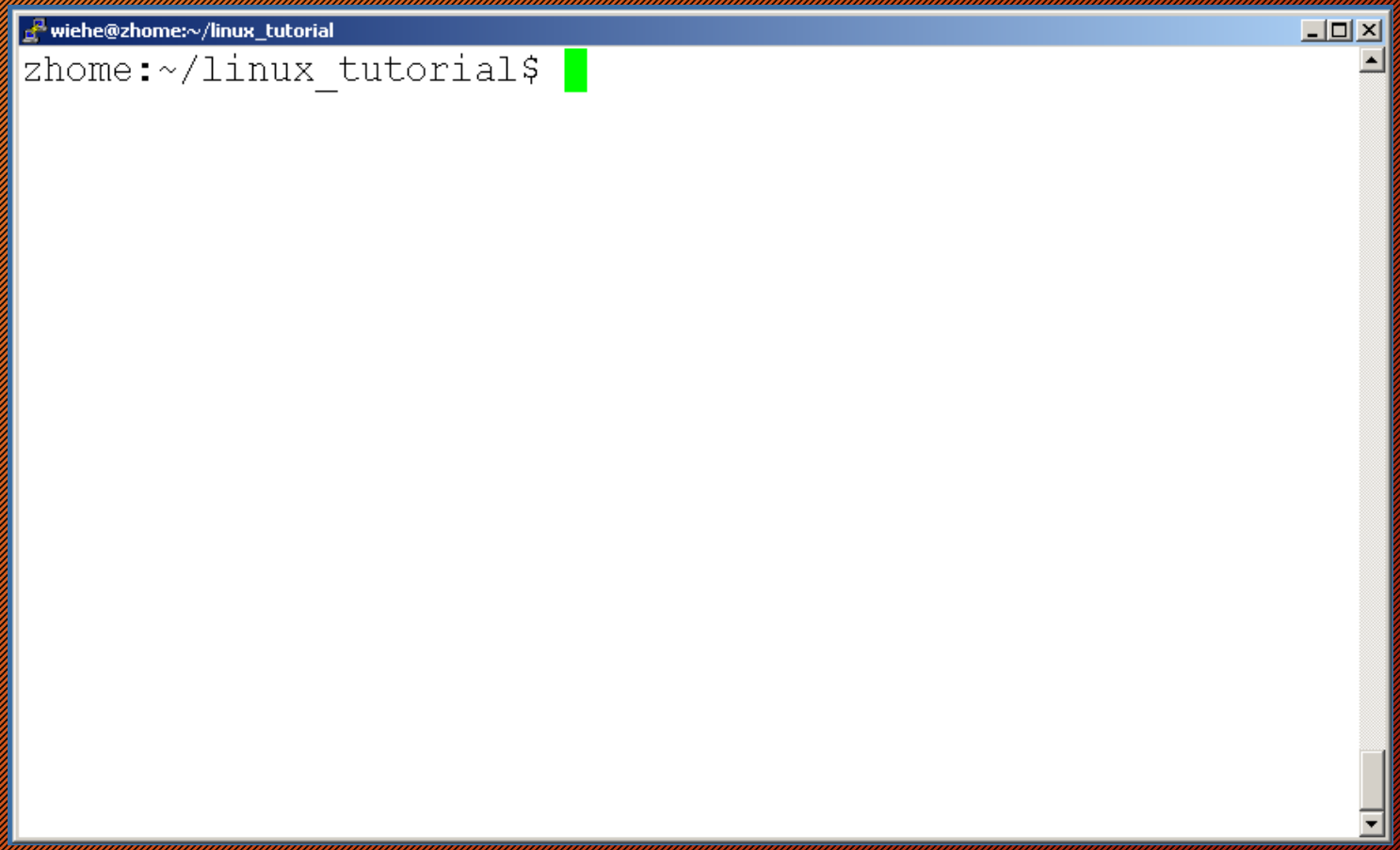
Linux Commands

CS3500 Operating System

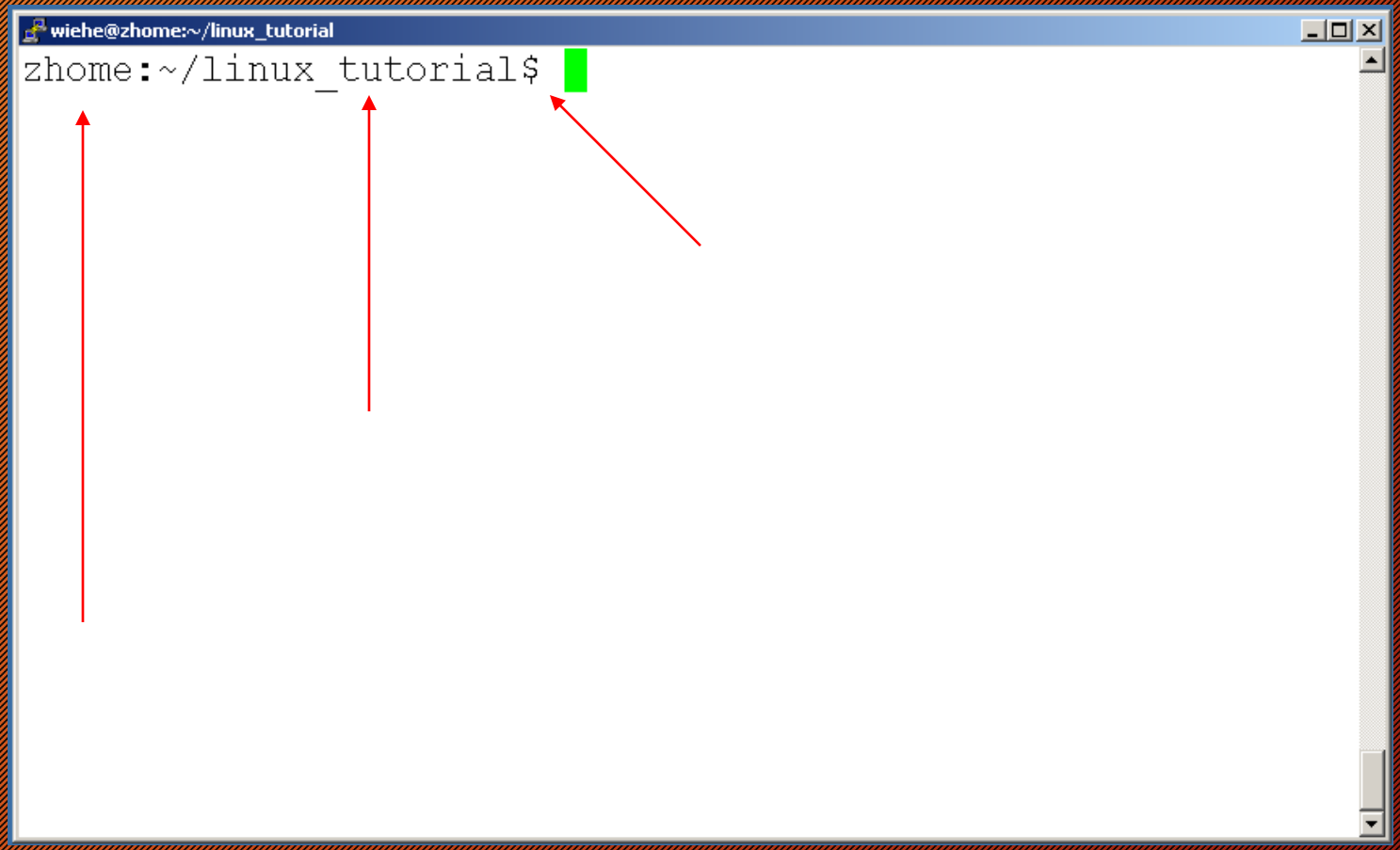
Connecting to a Unix/Linux system

- Open up a terminal:



Connecting to a Unix/Linux system

- Open up a terminal:



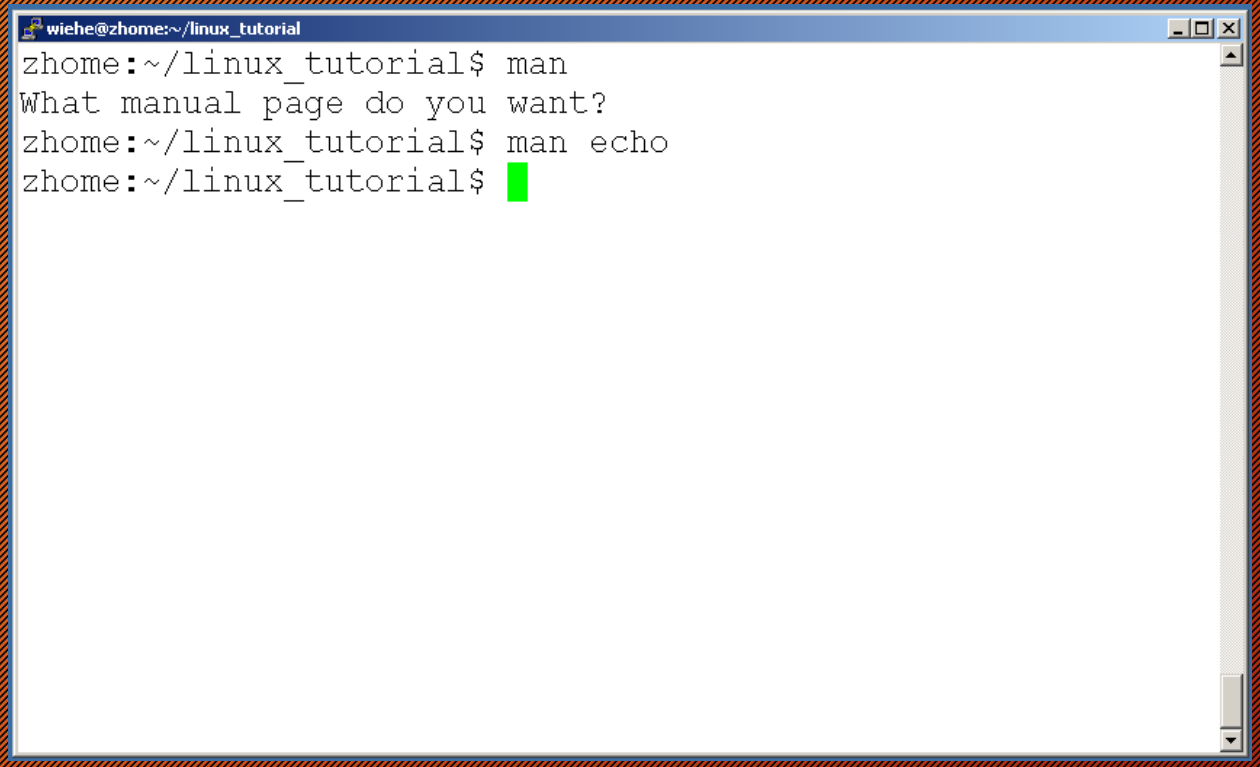
What exactly is a “shell”?

- 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!**

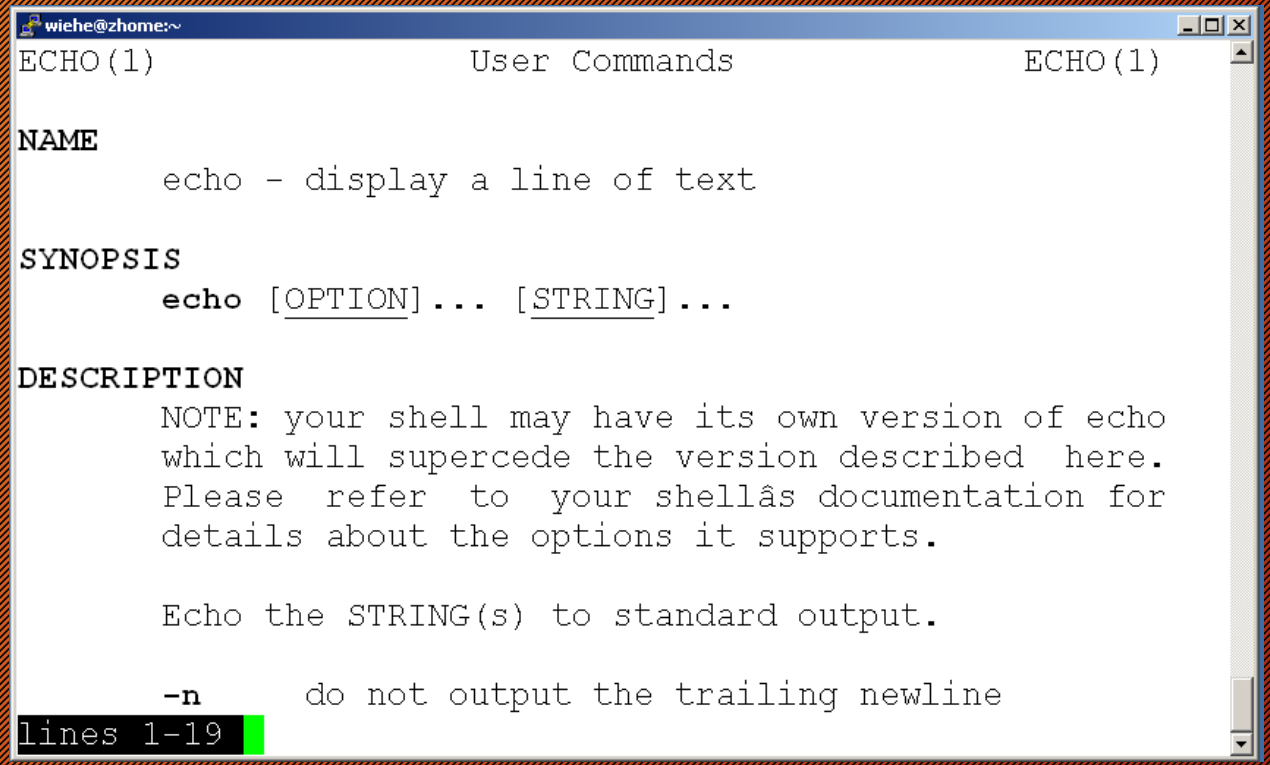
Help!

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

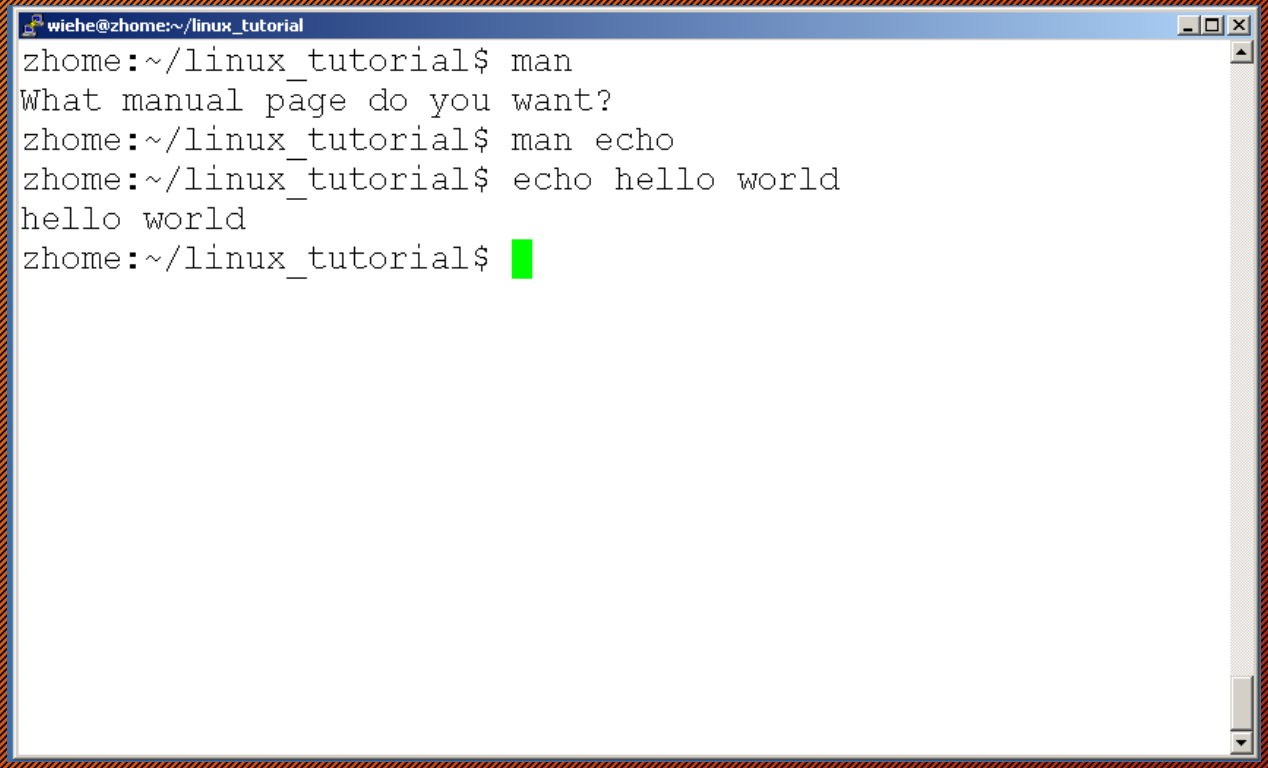
Help!


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

Help!

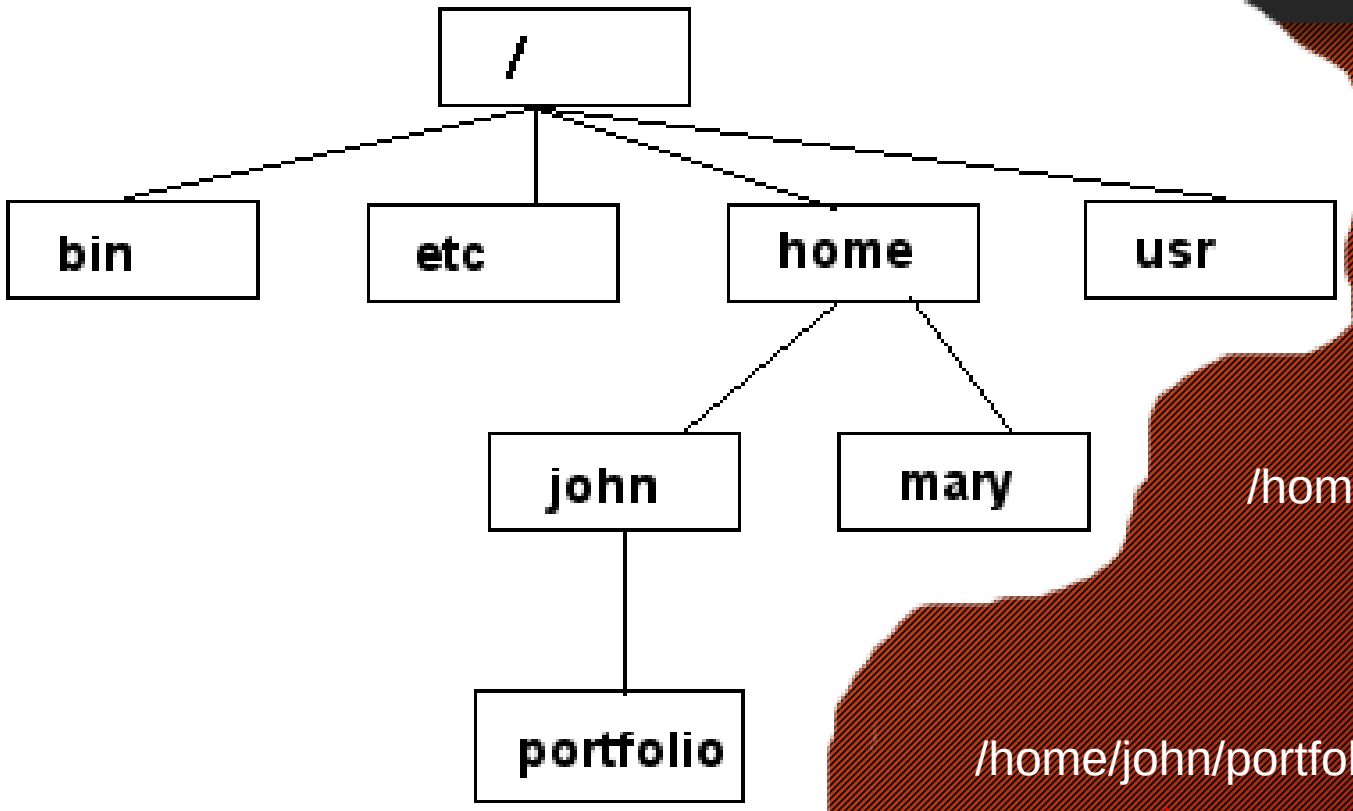

wiehe@zhome:~
ECHO(1) User Commands ECHO(1)
NAME
echo - display a line of text
SYNOPSIS
echo [OPTION]... [STRING]...
DESCRIPTION
NOTE: your shell may have its own version of echo which will supercede the version described here. Please refer to your shellâs documentation for details about the options 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!**



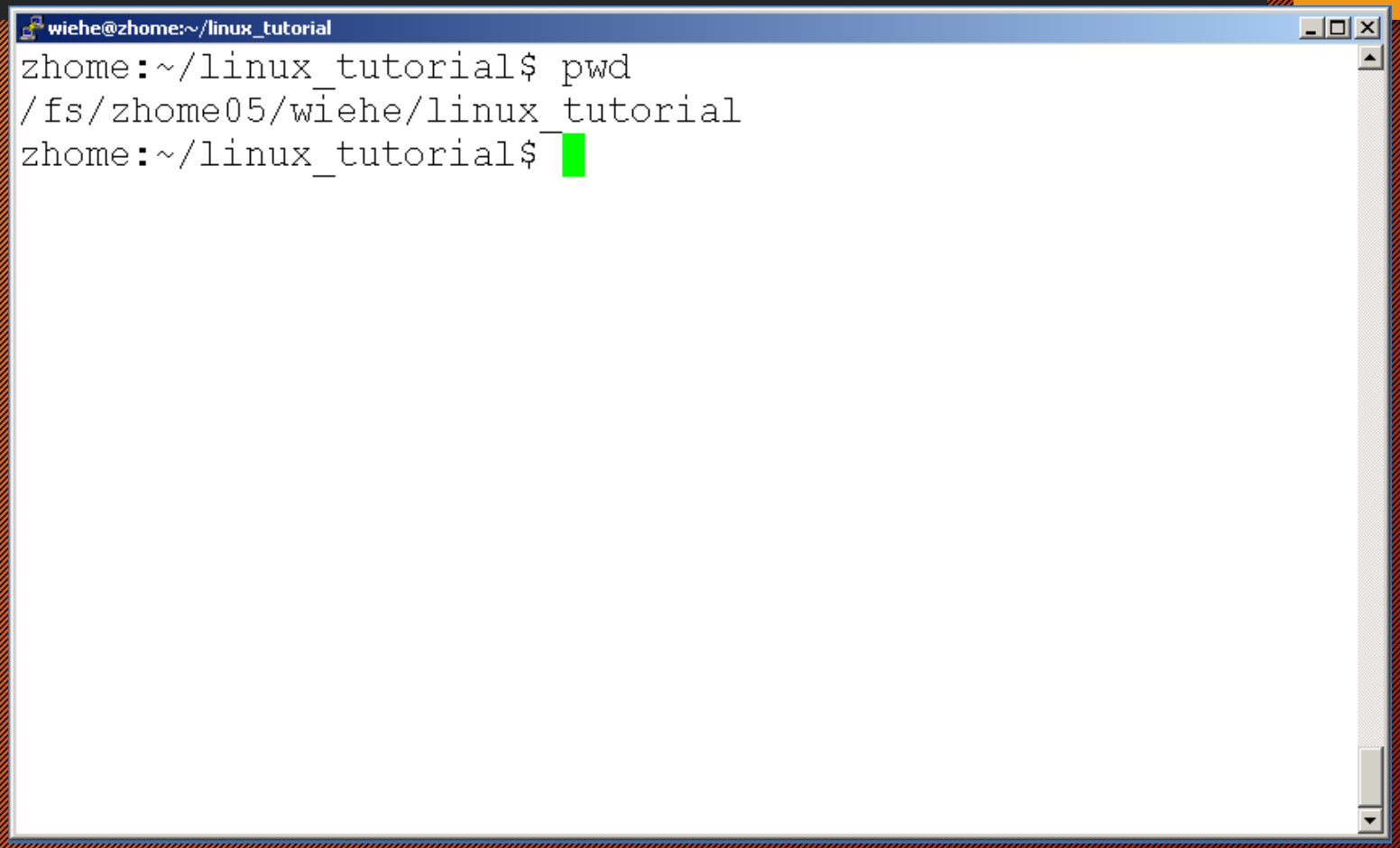
/home/mary/

/home/john/portfolio/

The Path

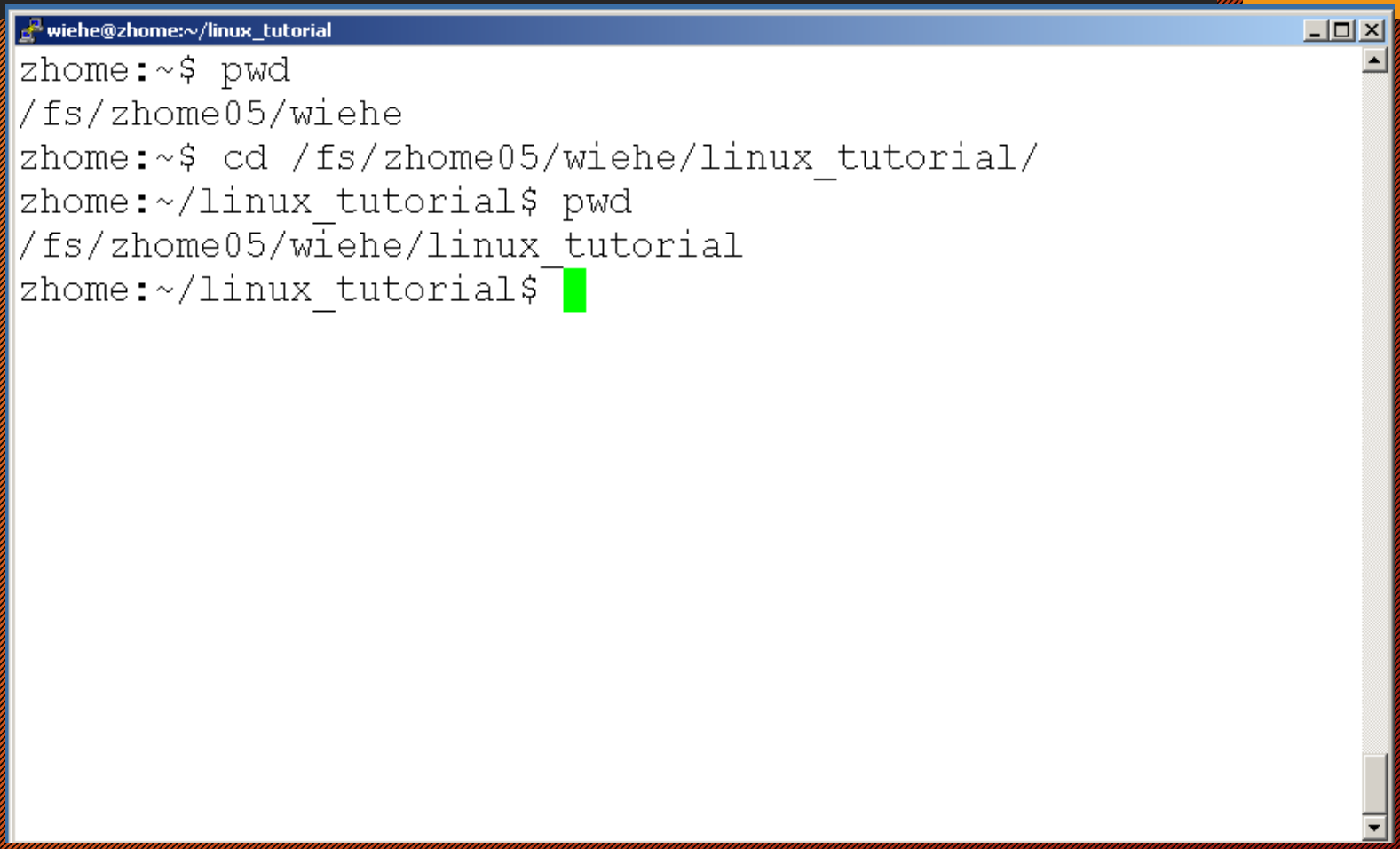
Command: pwd

- To find your current path use “pwd”



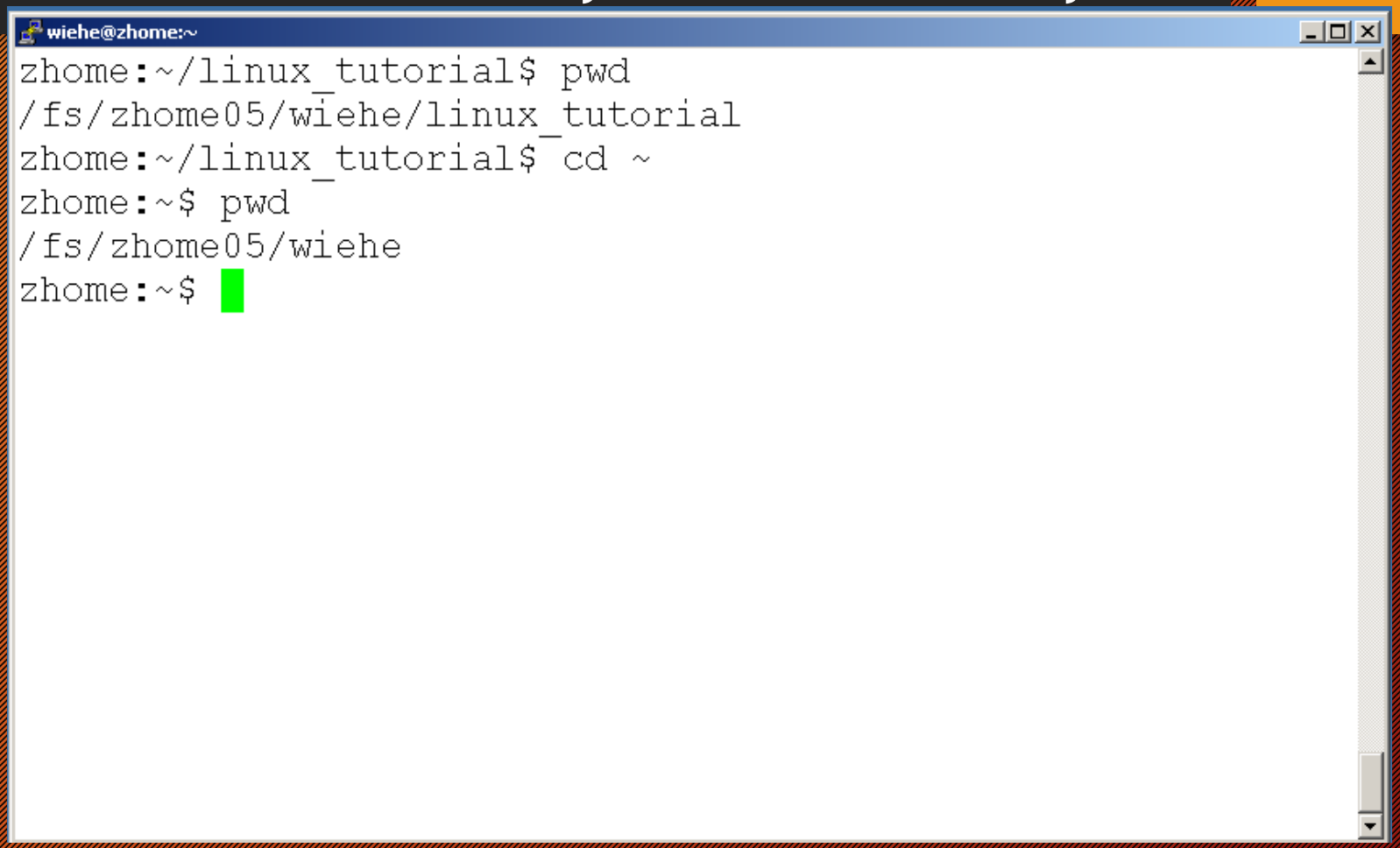
Command: cd

- To change to a specific directory use “cd”



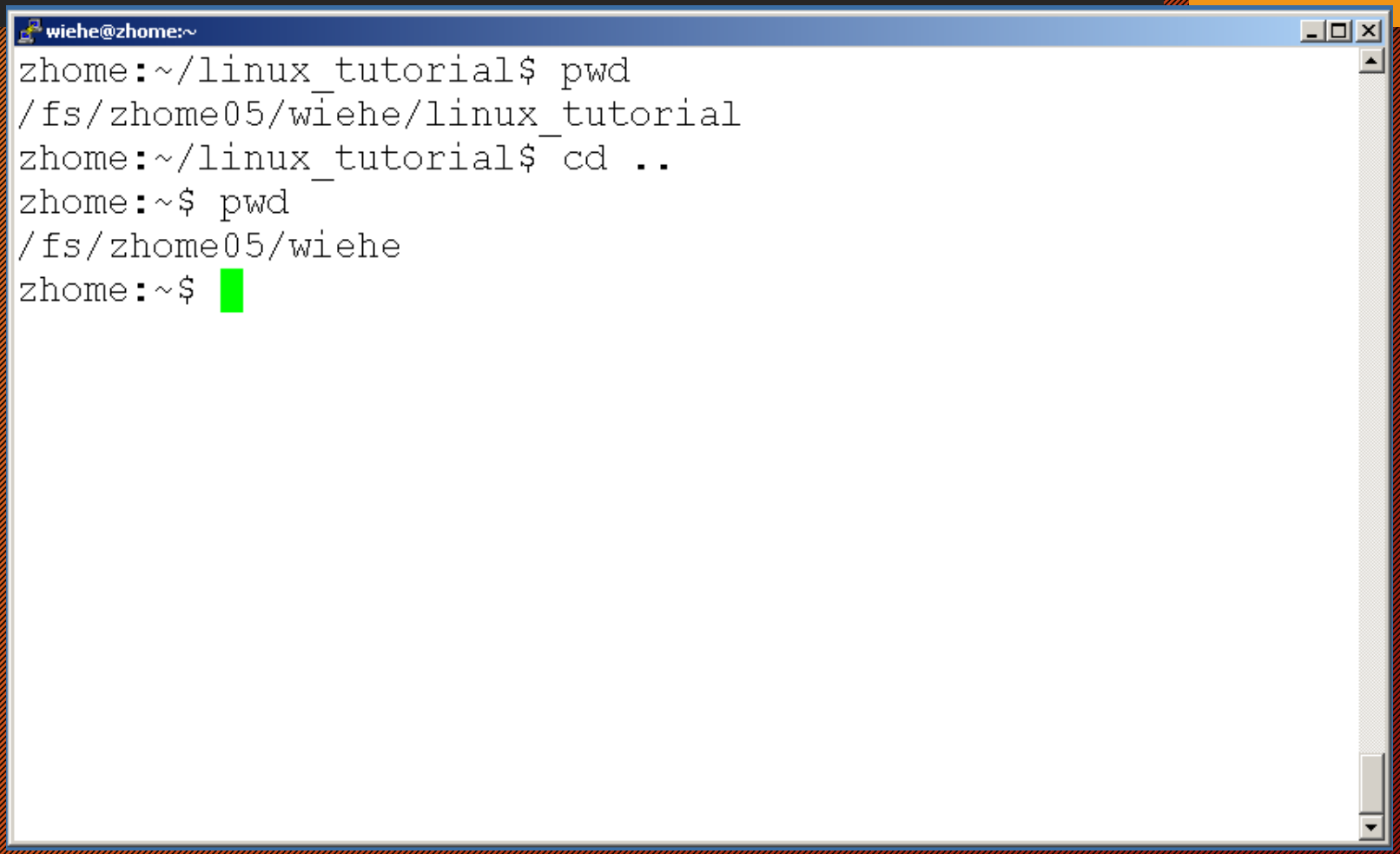
Command: cd

- “~” is the location of your home directory



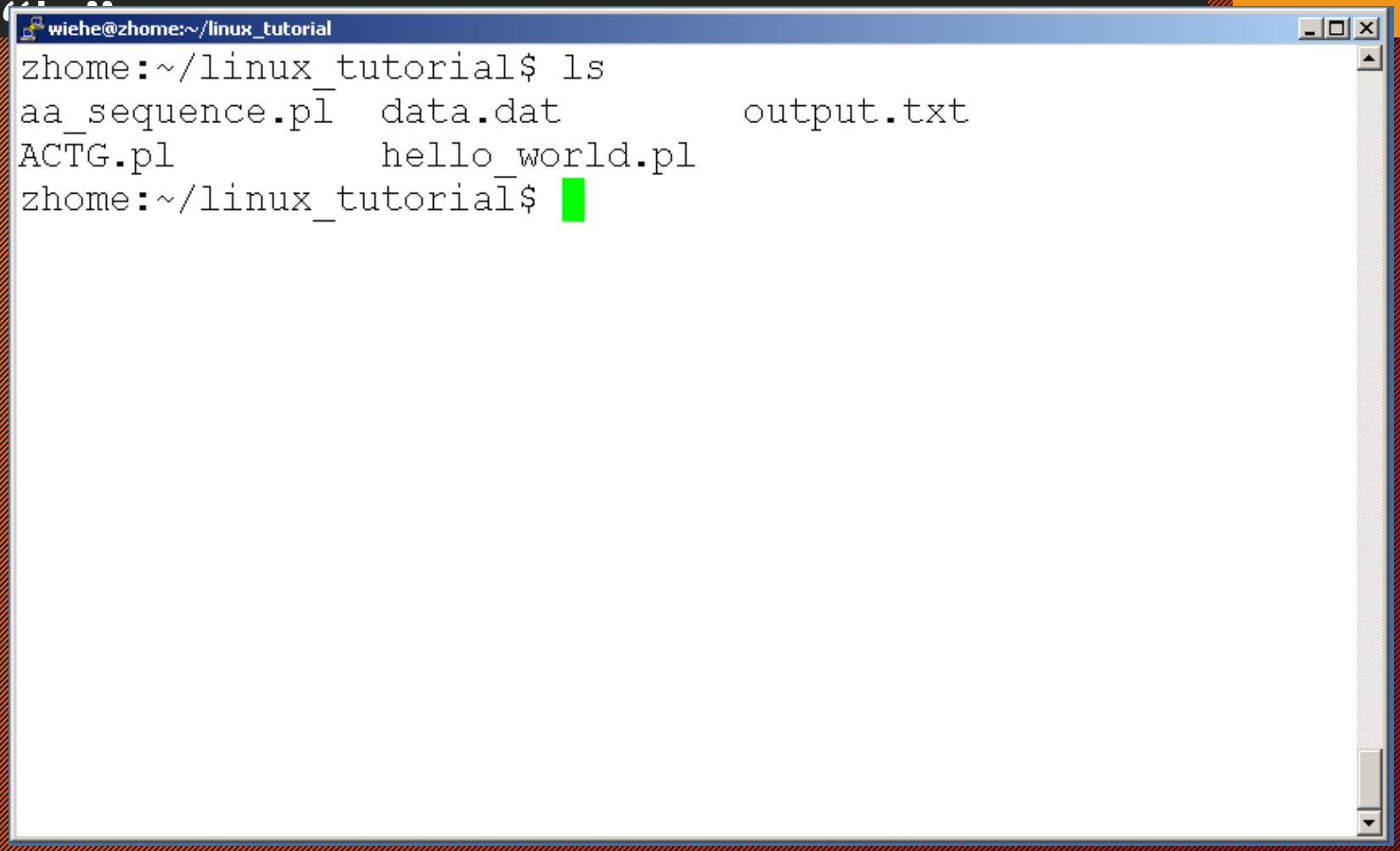
Command: cd

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



Command: ls

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

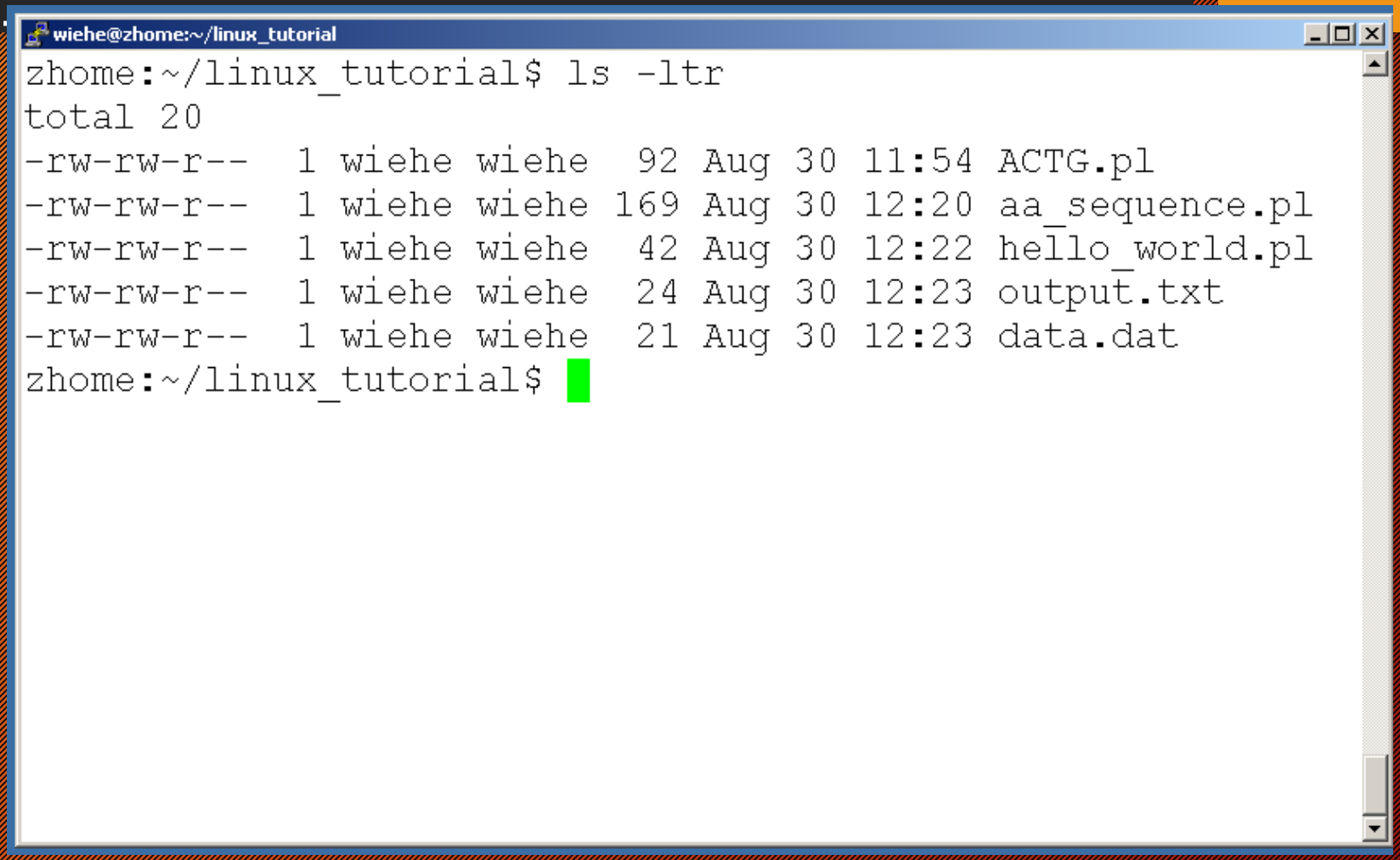


Command: ls

- 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"

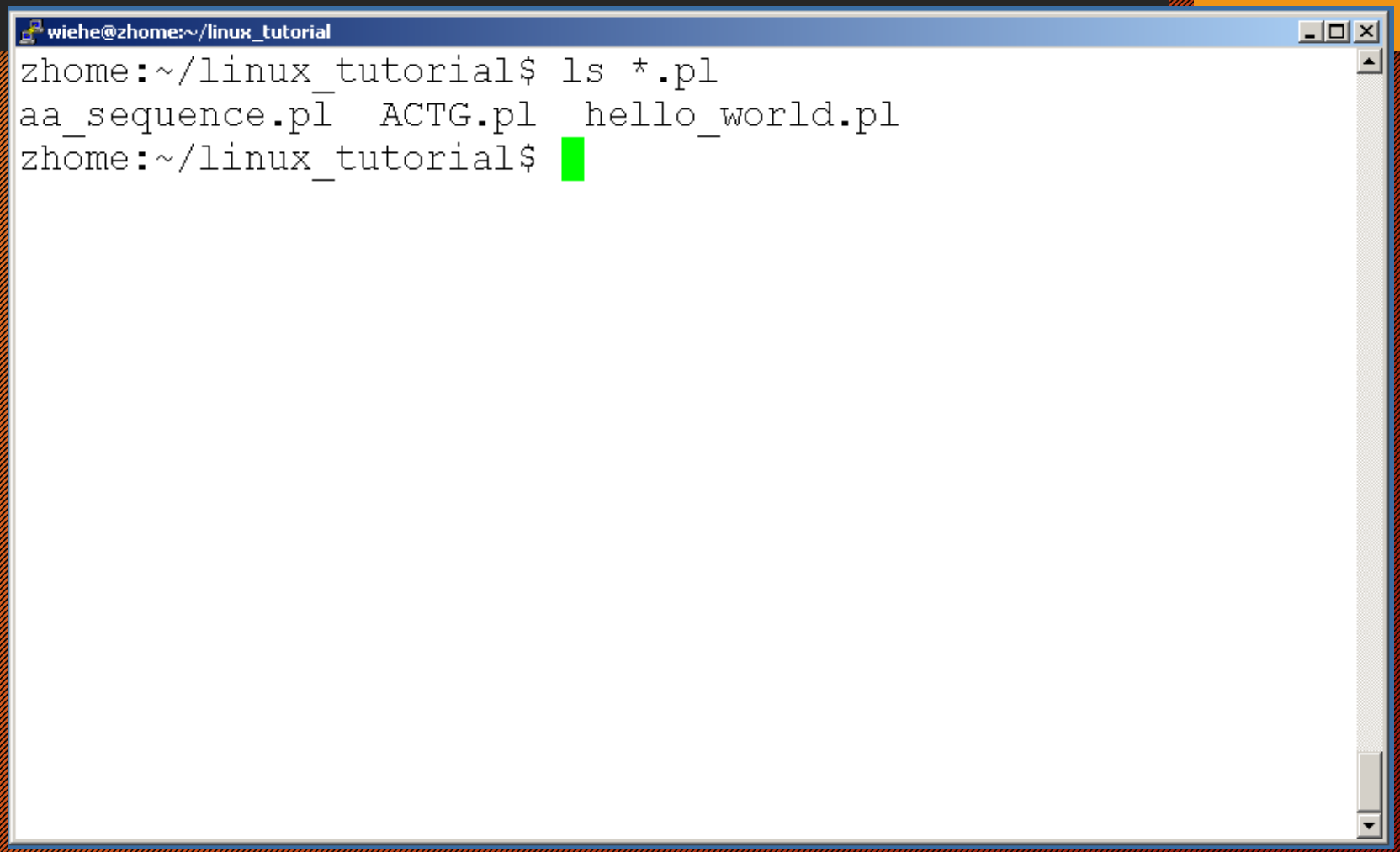
Command: ls -ltr

- List files by time in reverse order with long l



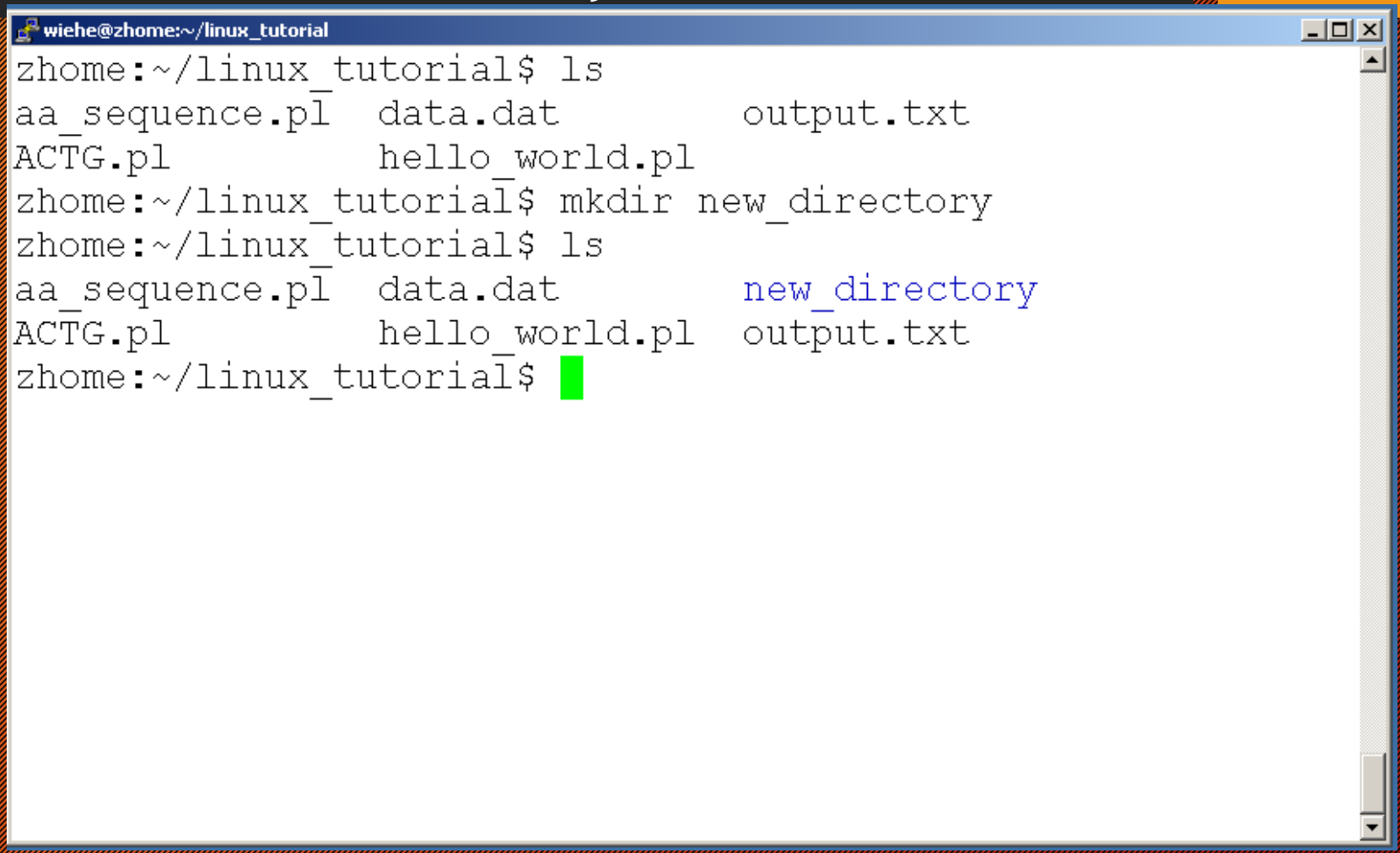
General Syntax: *

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



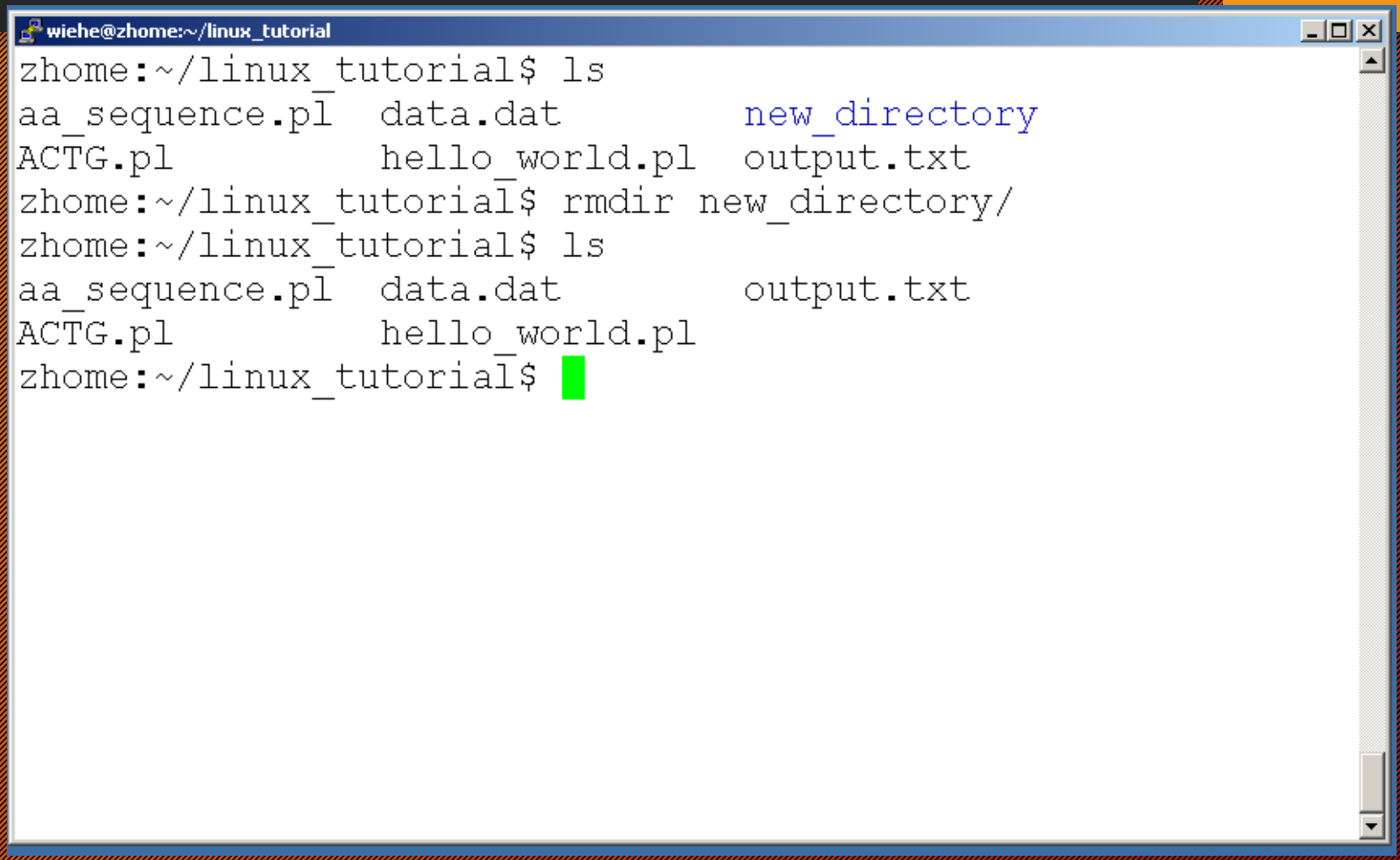
Command: mkdir

- To create a new directory use “mkdir”



Command: rmdir

- To remove and empty directory use “rmdir”



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

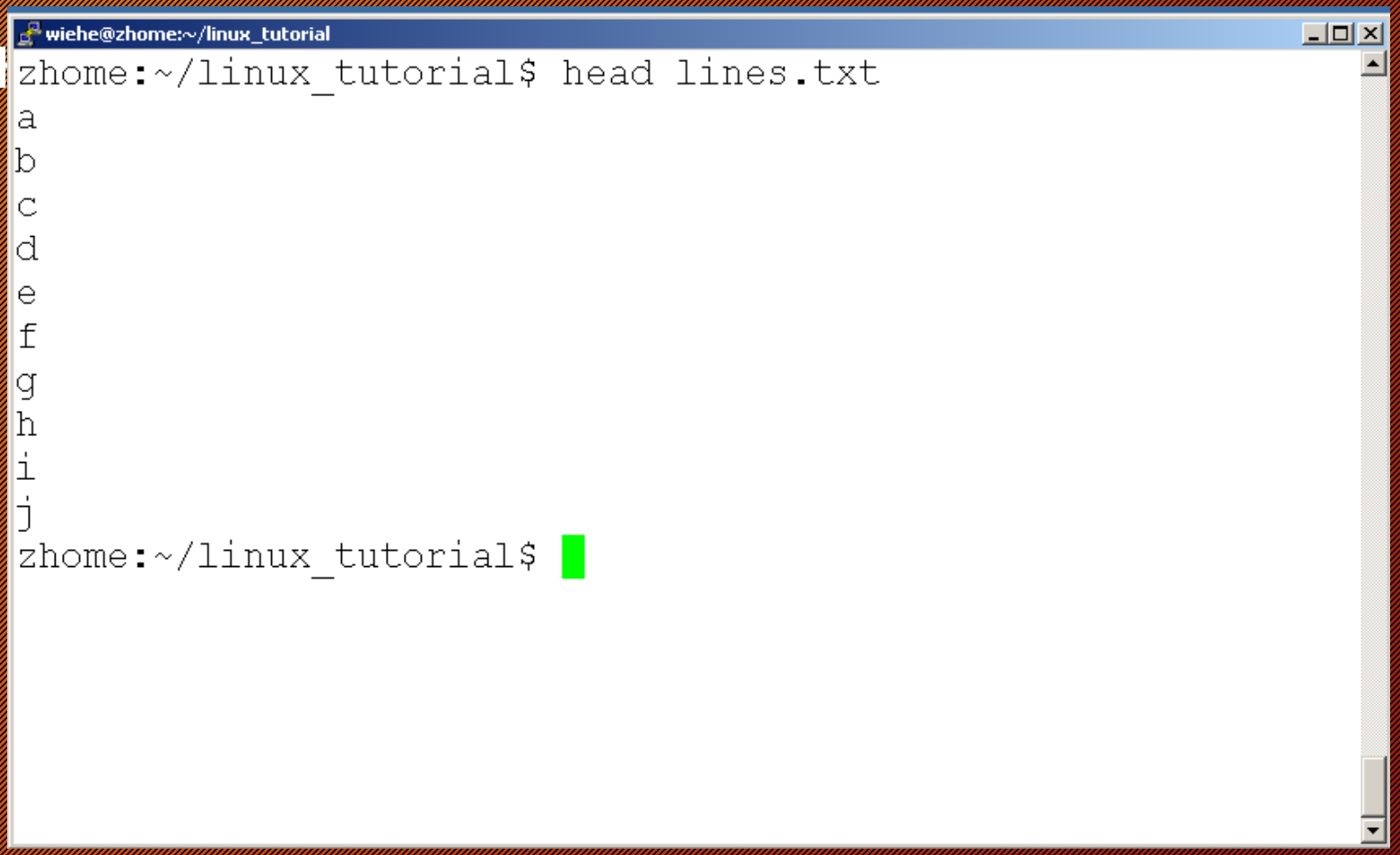
Command: less

- “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

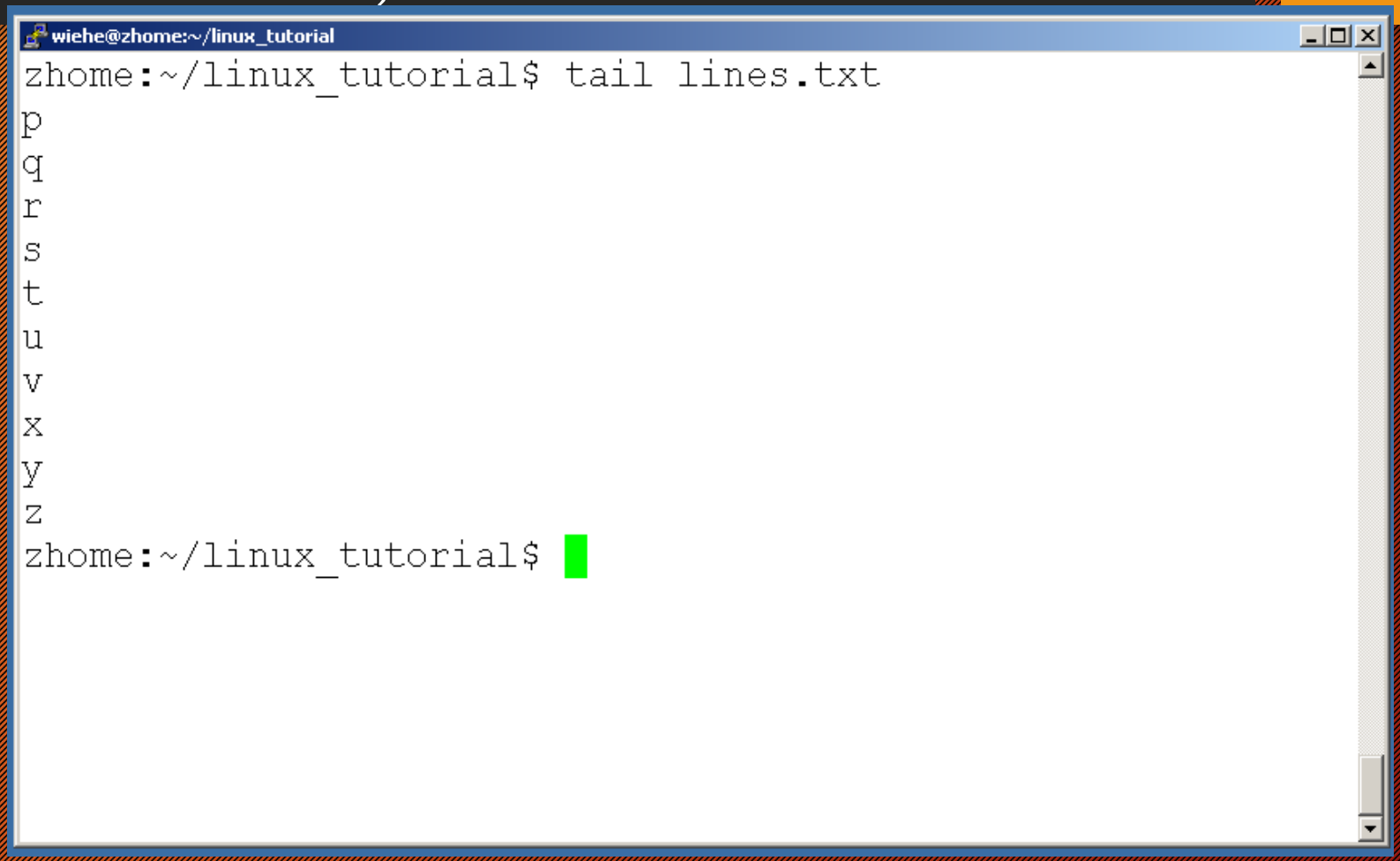
- “head” displays the top part of a file
- By default it shows the first 10 lines
- -n option allows you to change that
- “head -n50 file.txt” displays the first 50 lines of file.txt

Command: head

- 

Command: tail

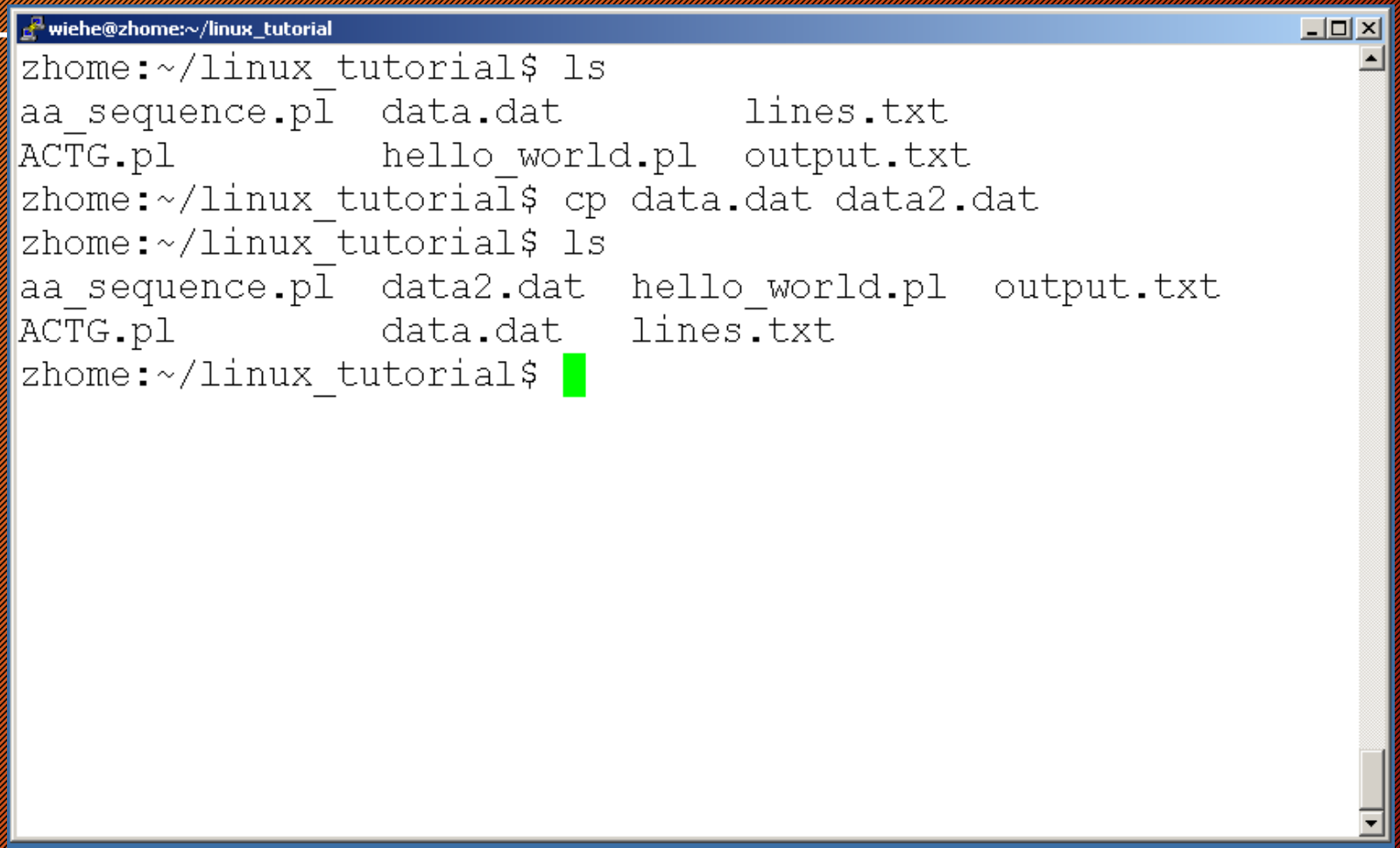
- Same as head, but shows the last lines



File Commands

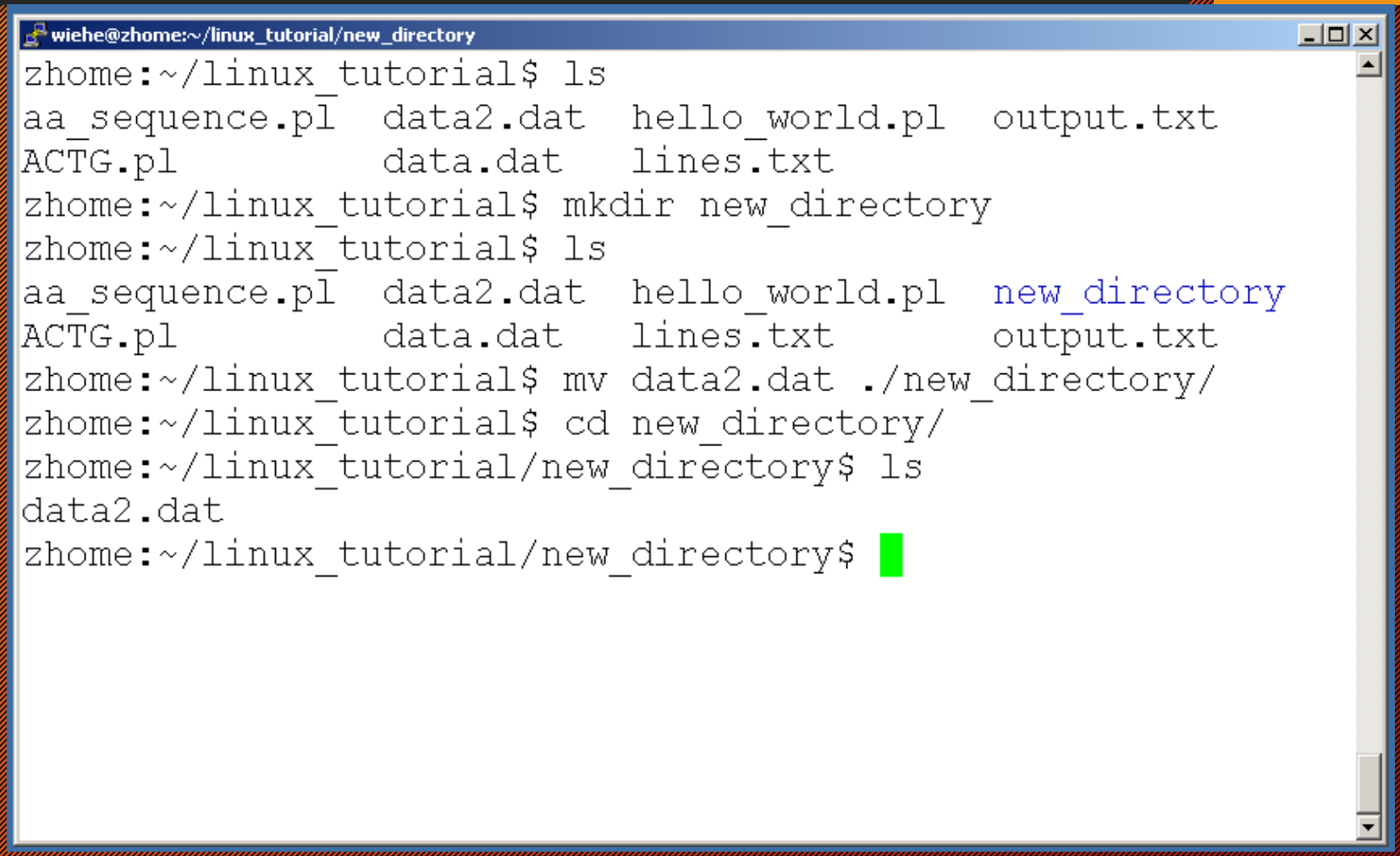
- Copying a file: cp
- Move or rename a file: mv
- Remove a file: rm

Command: cp

- 
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



Command: mv

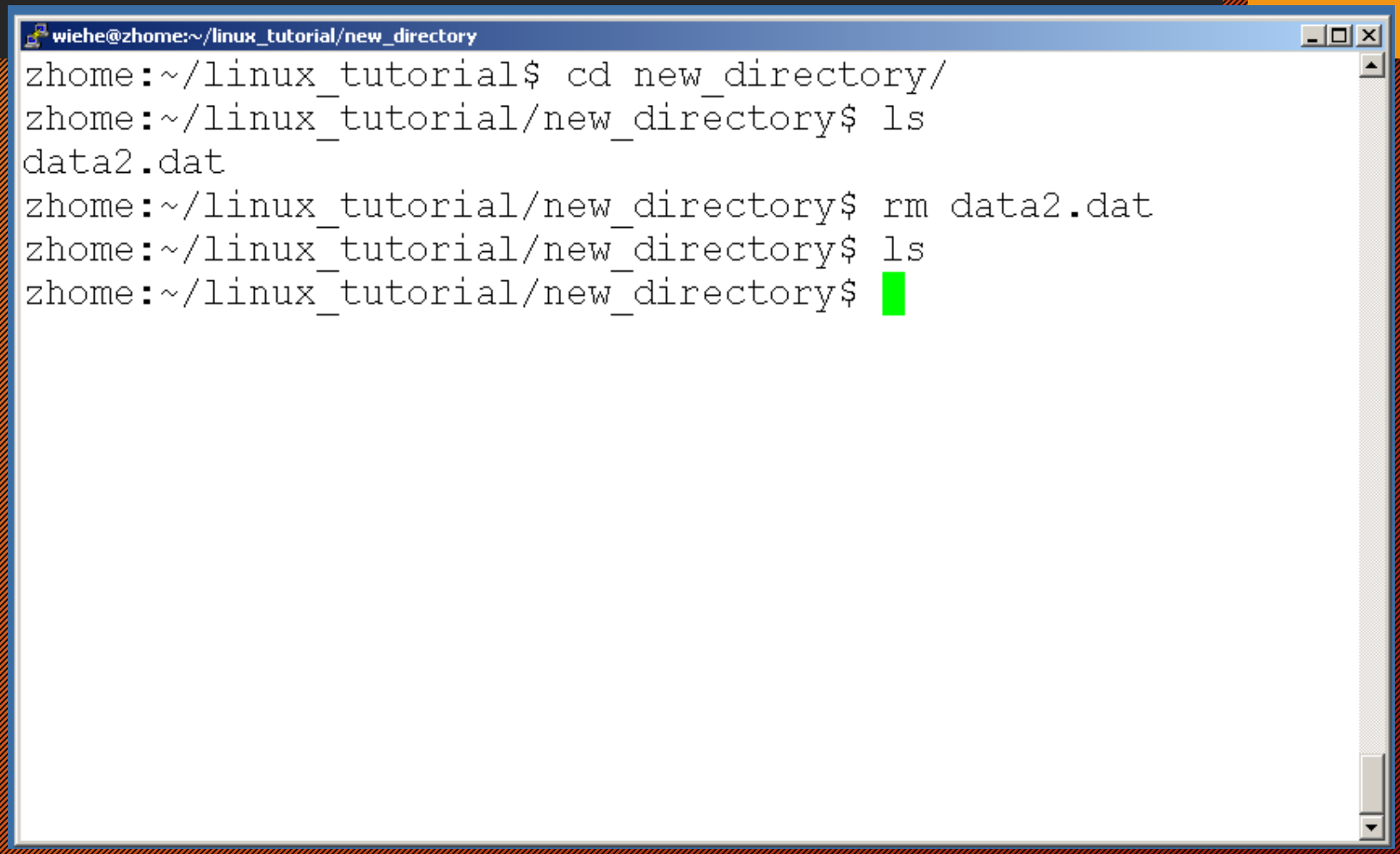
- m

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”



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

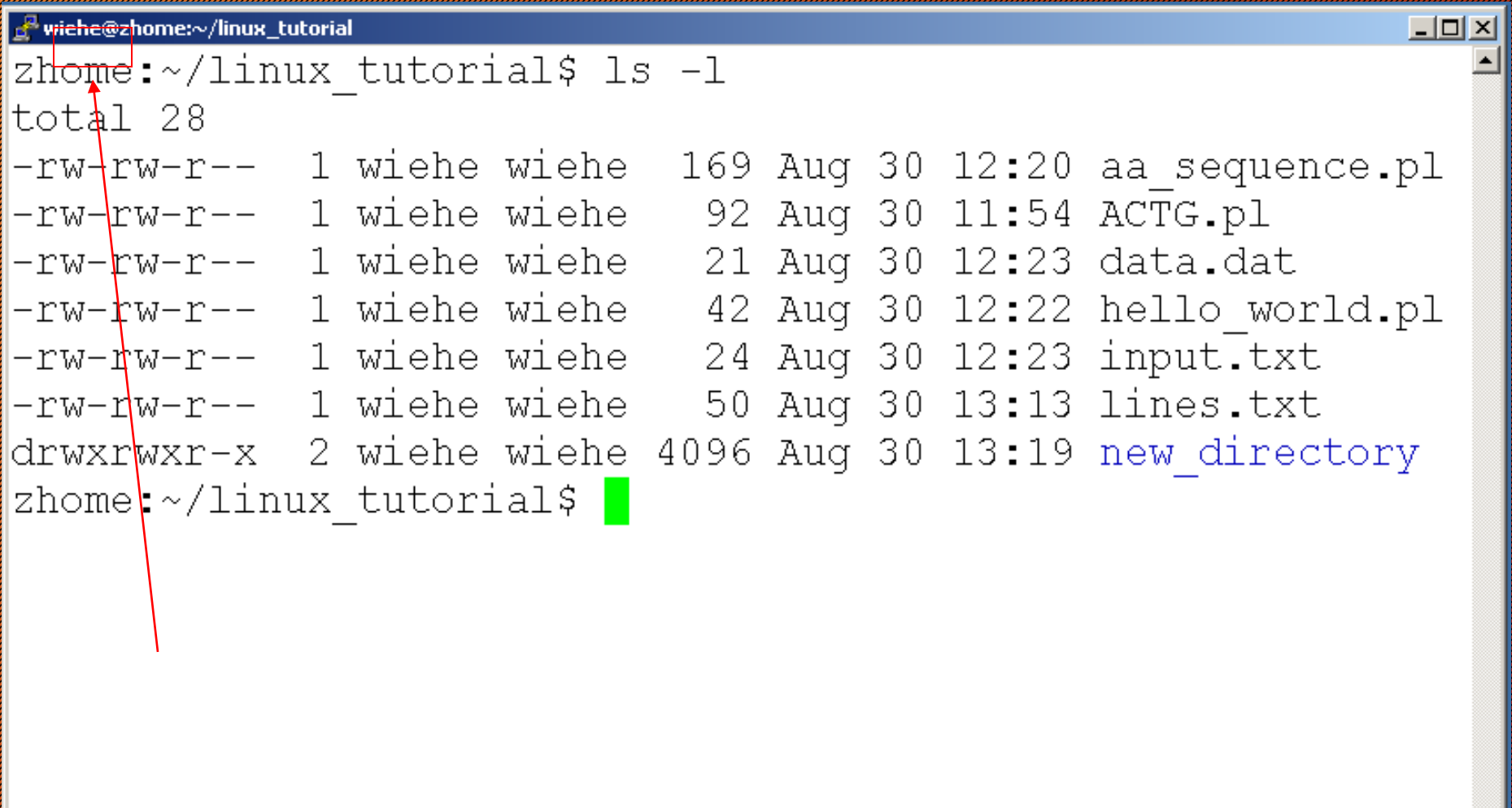
File permissions

- 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

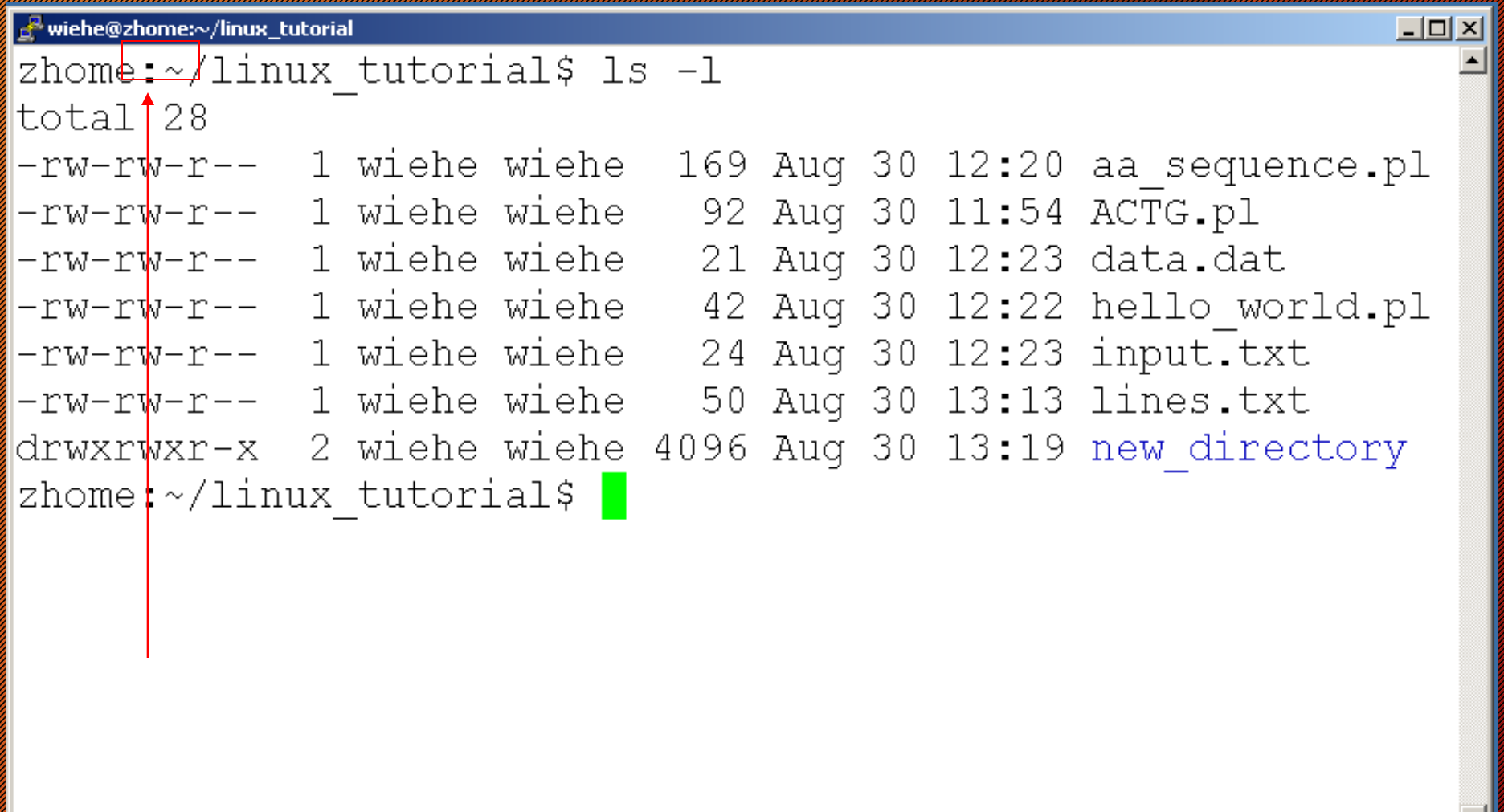
Permission levels

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

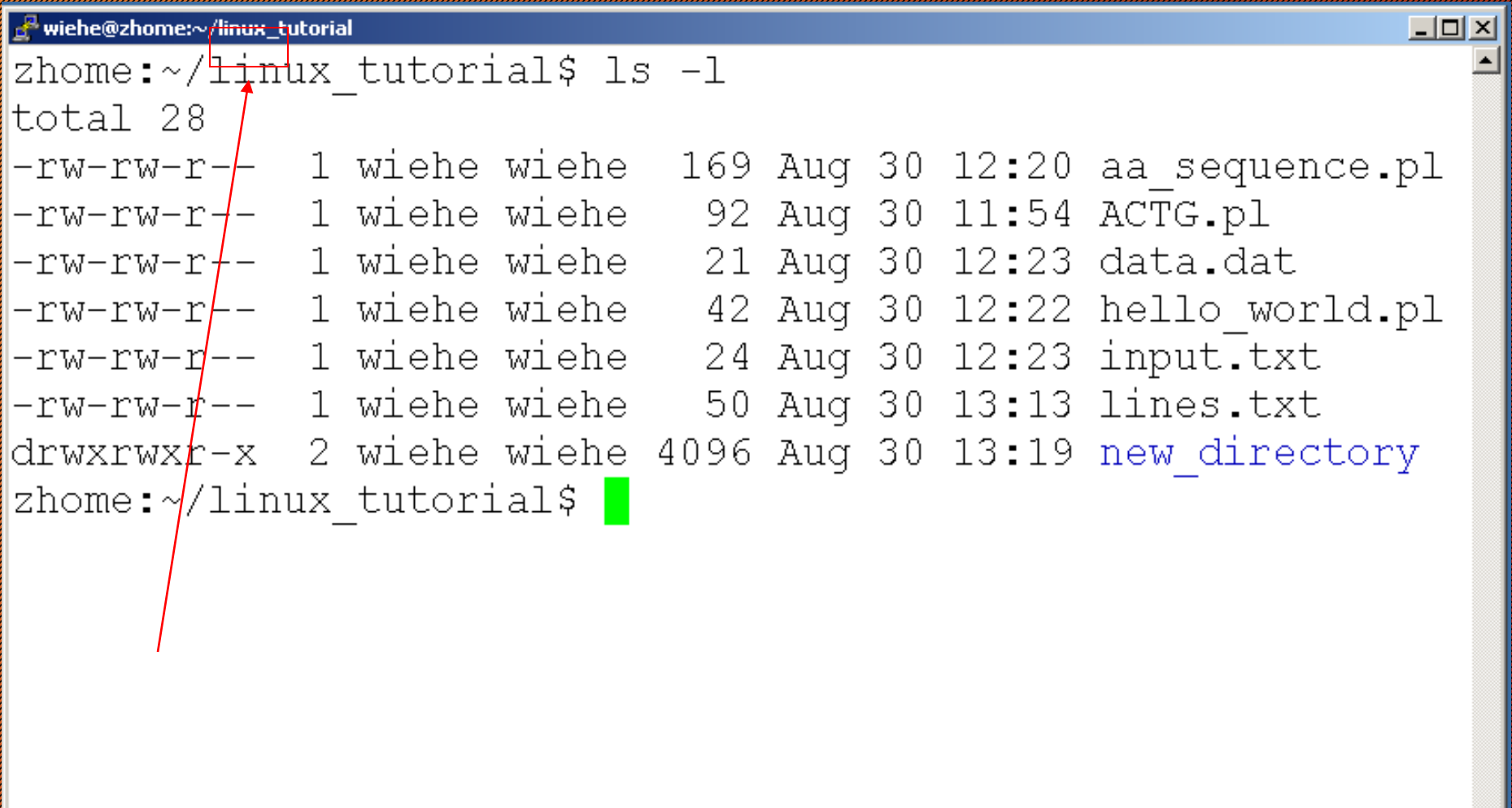
File Permissions



File Permissions

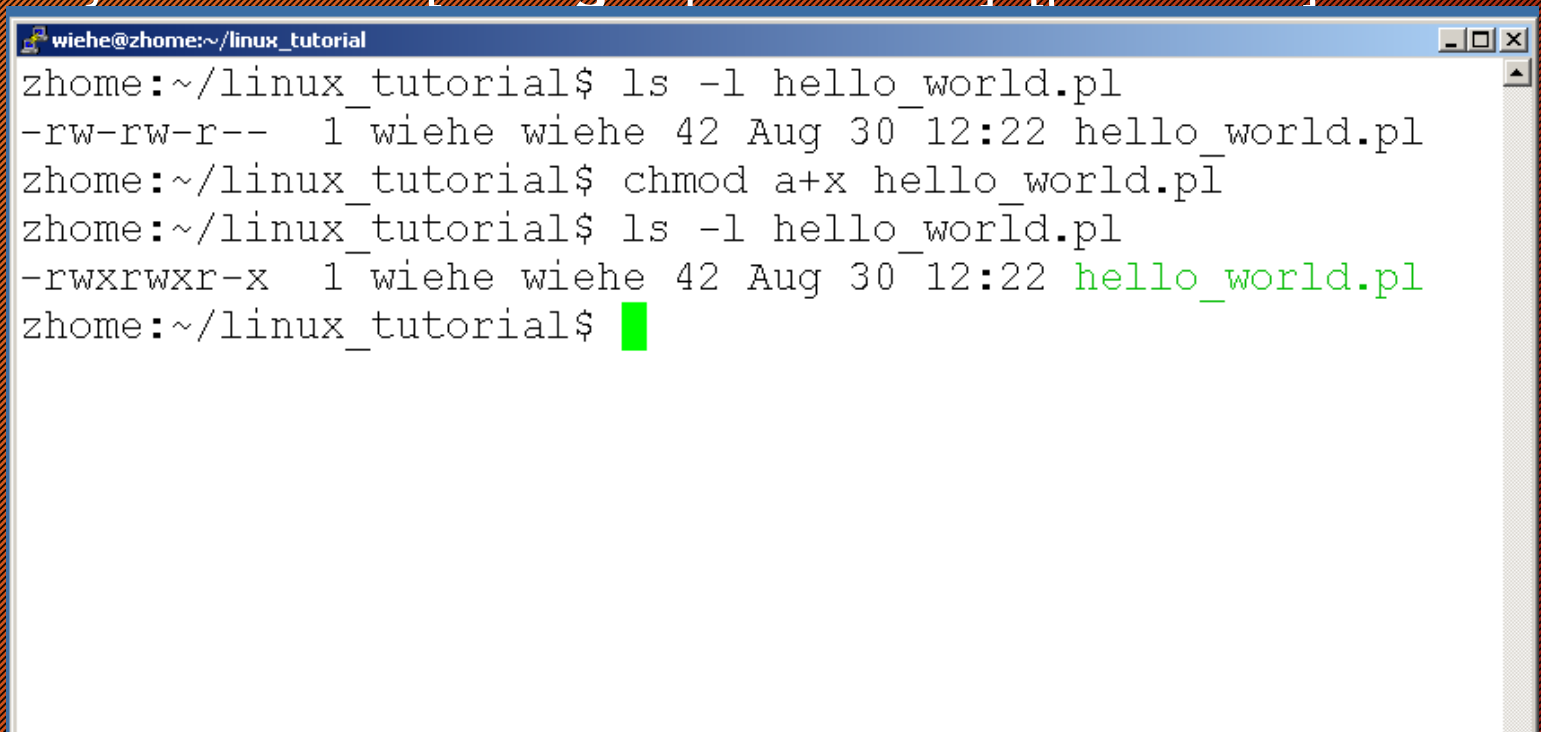


File Permissions



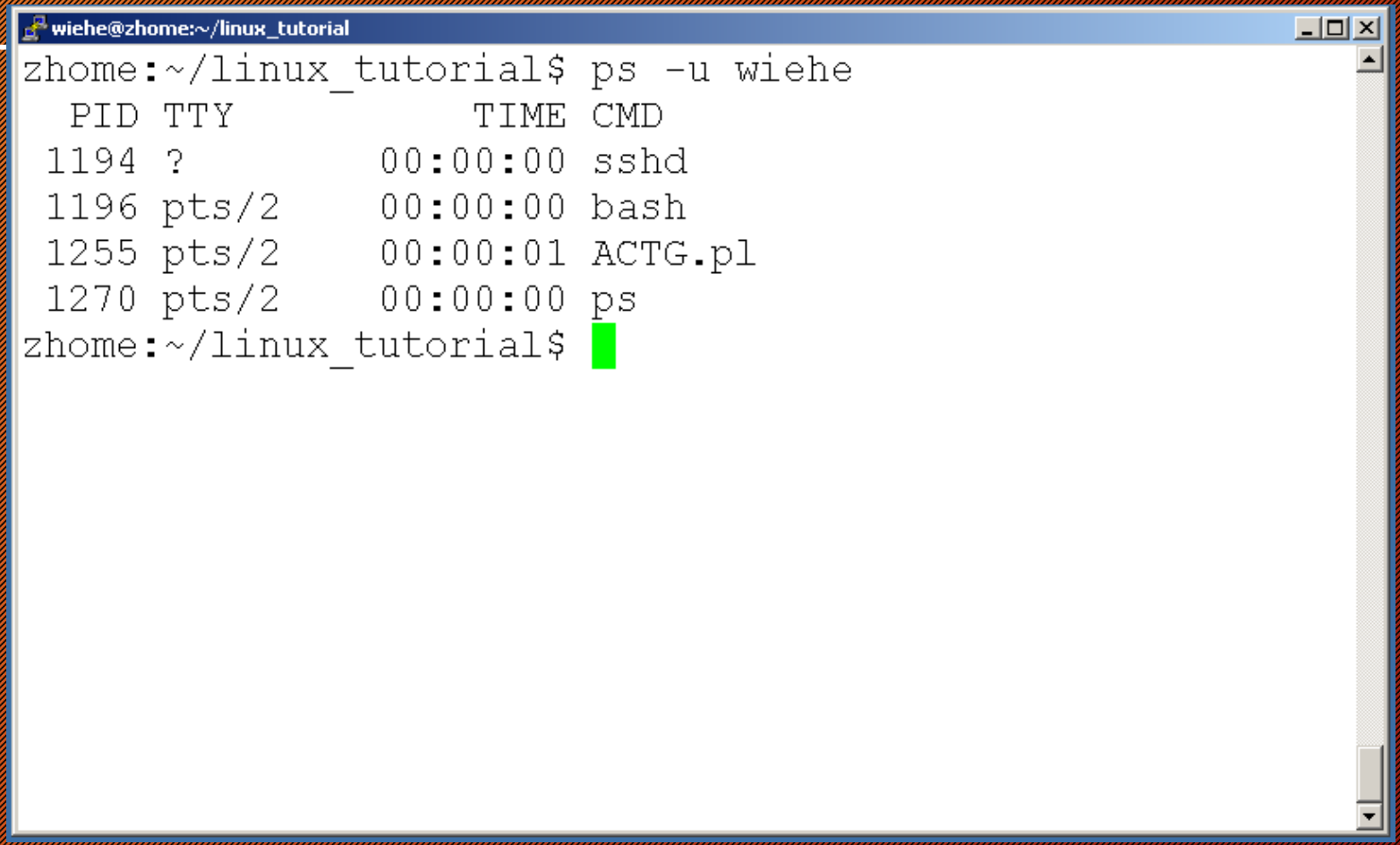
Command: chmod

- If you own the file, you can change it's permissions with "chmod"
  - Syntax: chmod [**u**ser/**g**roup/**o**thers/**a**ll]+[permission]



Command: ps

- T


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

-

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

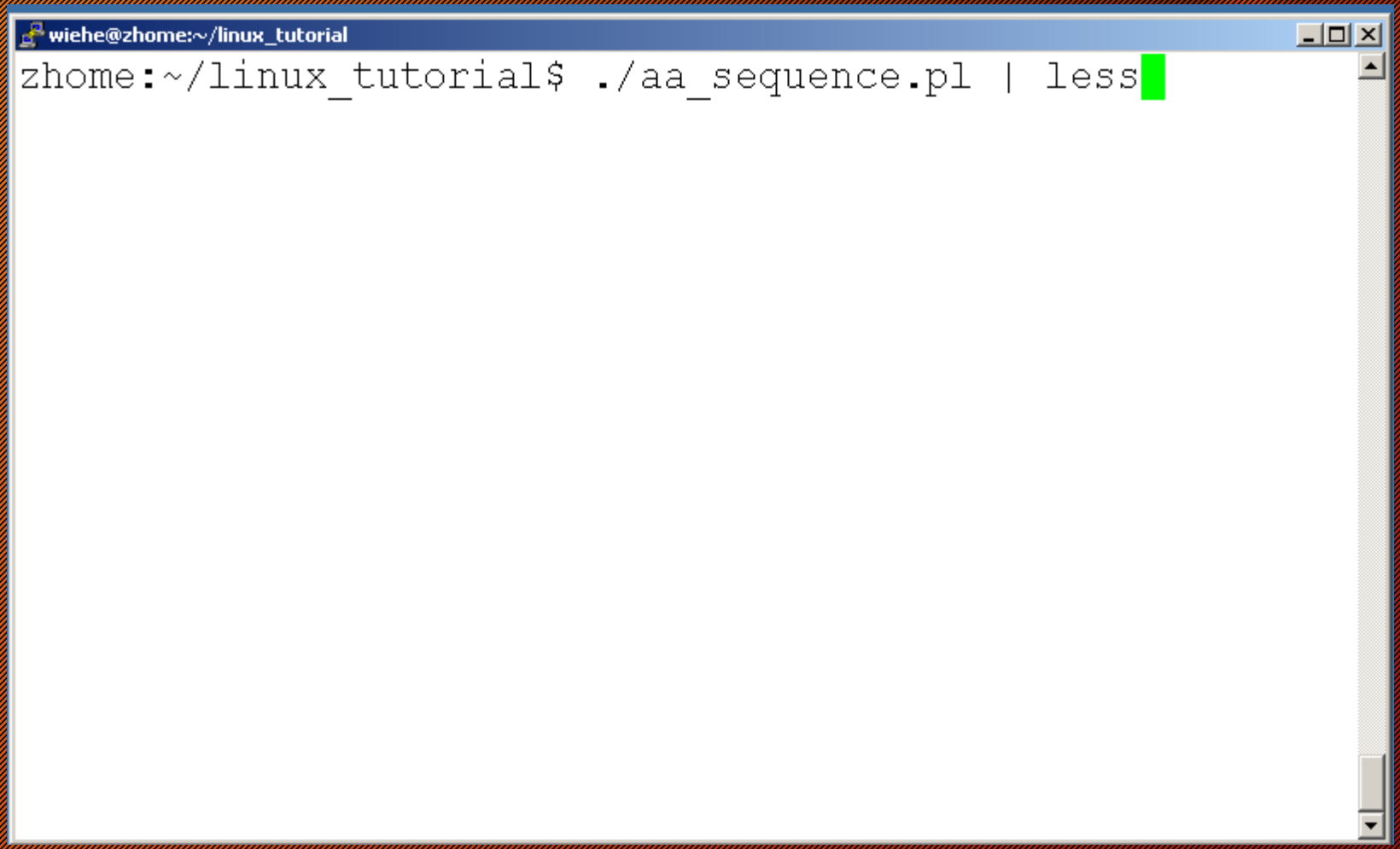
-

```
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$
```

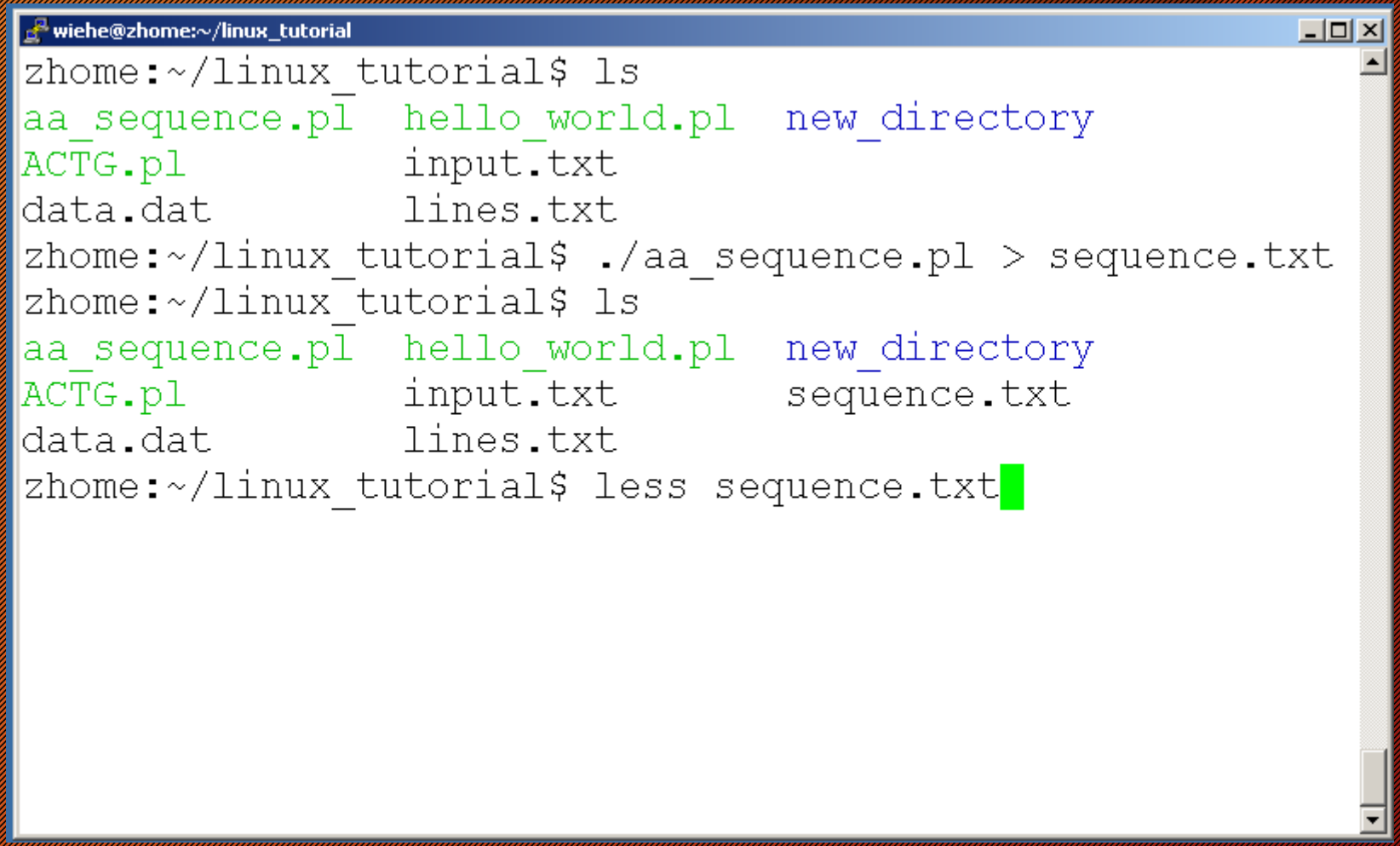
Input/Output Redirection ("piping")

- 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



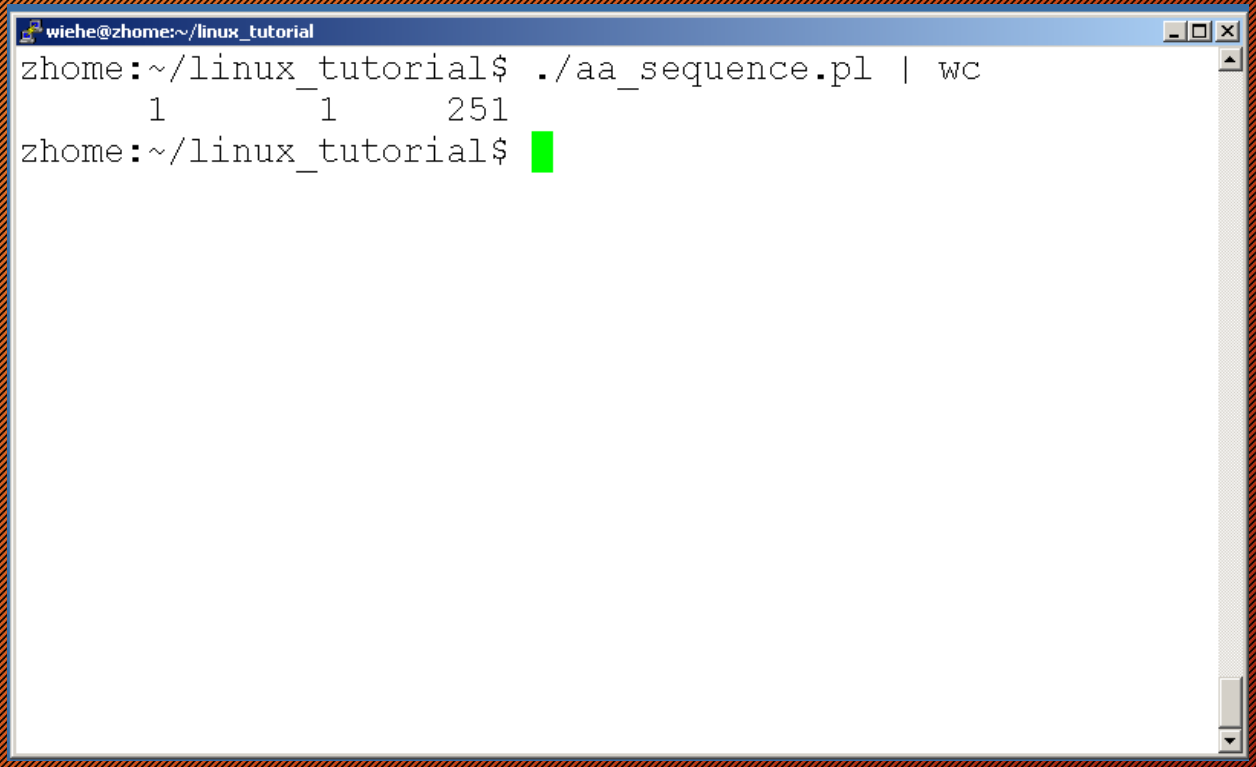
A few examples of piping



Command: wc

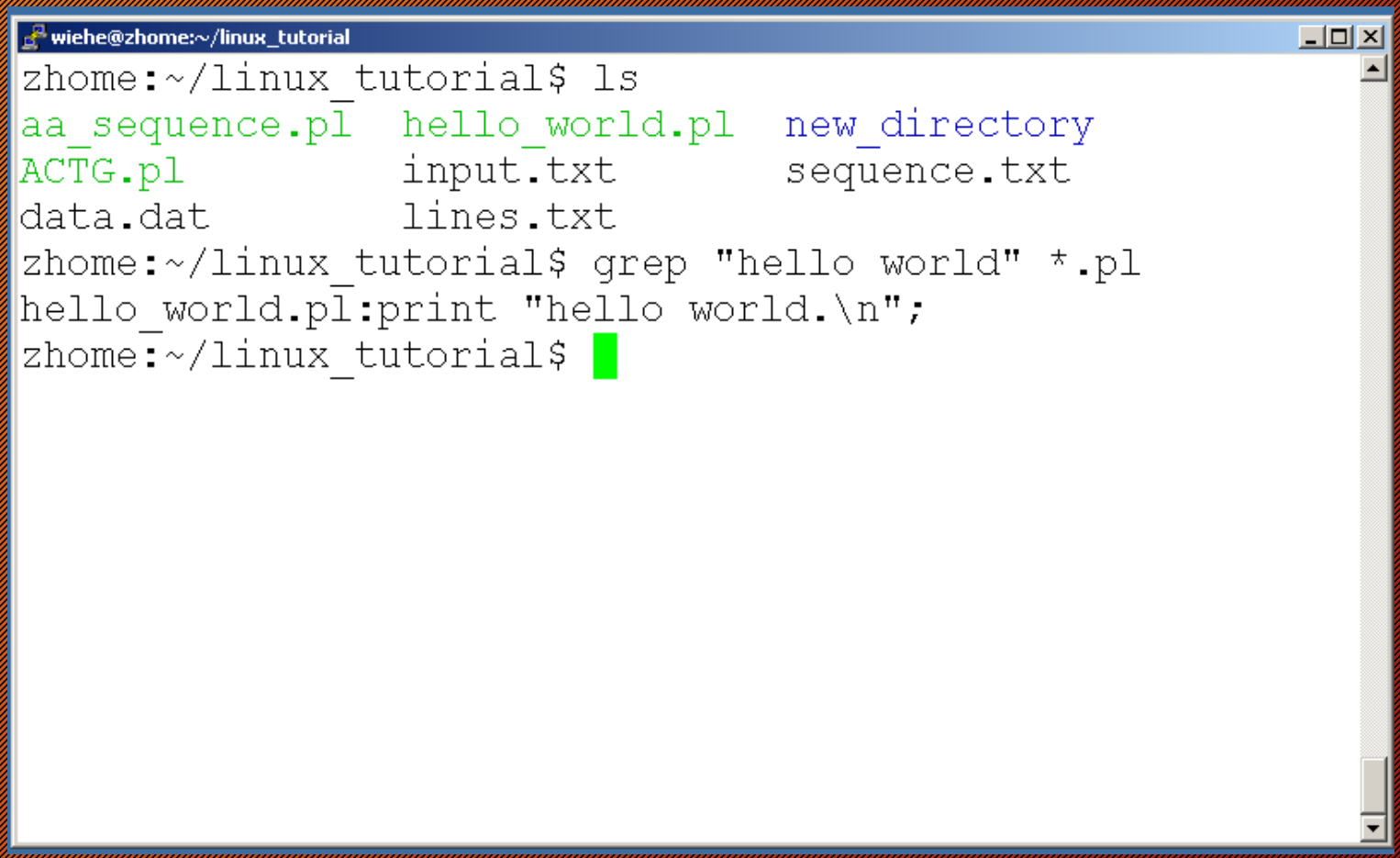
- To count the characters, words, and lines in a file use “wc”
- The first column in the output is lines, the second is words, and the last is characters

A few examples of piping



Command: grep

- To search files in a directory for a specific string use "grep"



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