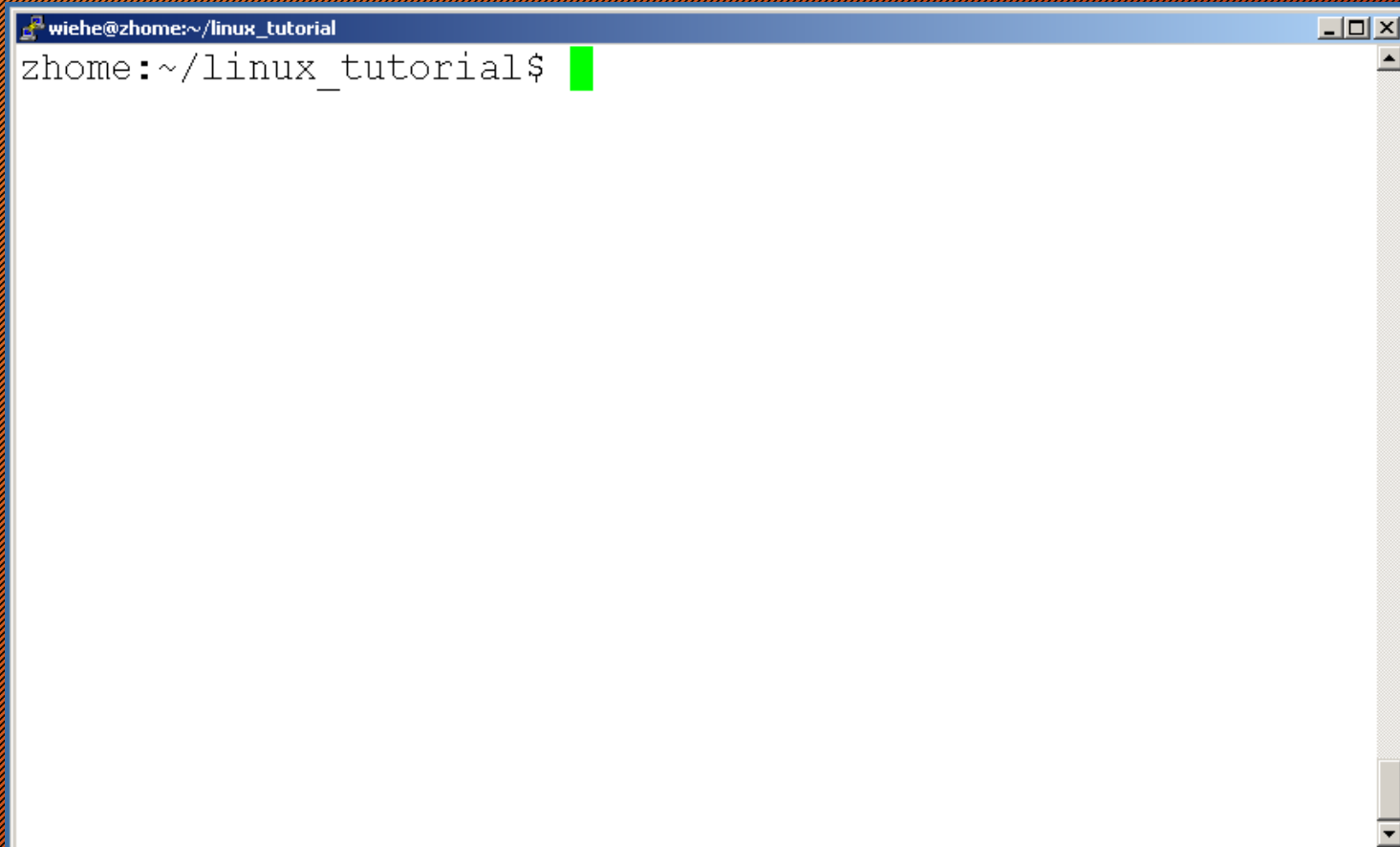


Linux Commands

CS3500 Operating System

Connecting to a Unix/Linux system

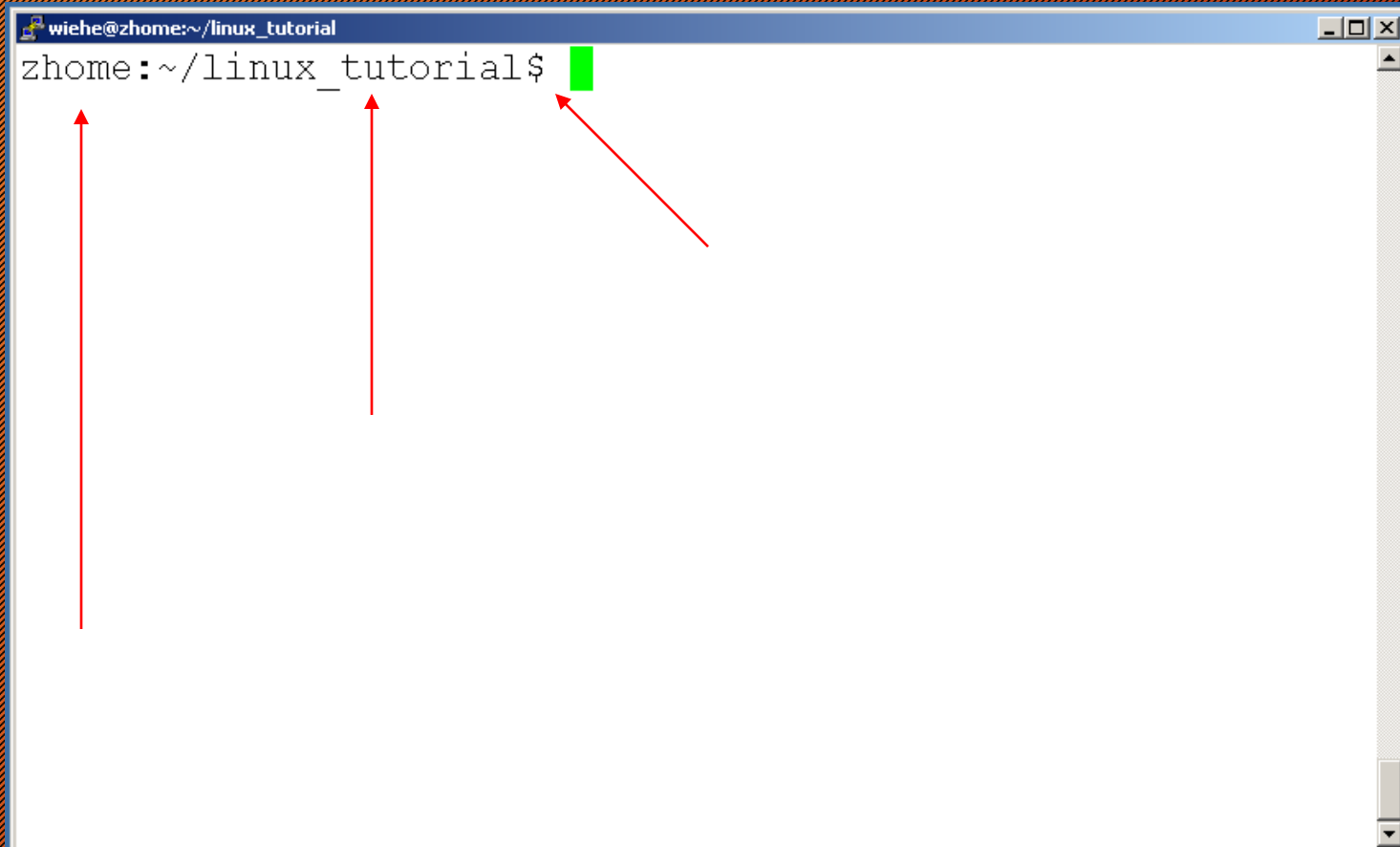
- Open up a terminal:

A screenshot of a terminal window. The title bar at the top reads 'wiehe@zhome:~/linux_tutorial'. The terminal content shows the prompt 'zhome:~/linux_tutorial\$' followed by a green cursor. 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:



A screenshot of a terminal window titled "wiehe@zhome:~/linux_tutorial". The prompt is "zhome:~/linux_tutorial\$". A green cursor is positioned at the end of the prompt. Three red arrows point to the prompt: one to the username "wiehe", one to the host "zhome", and one to the green cursor.

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

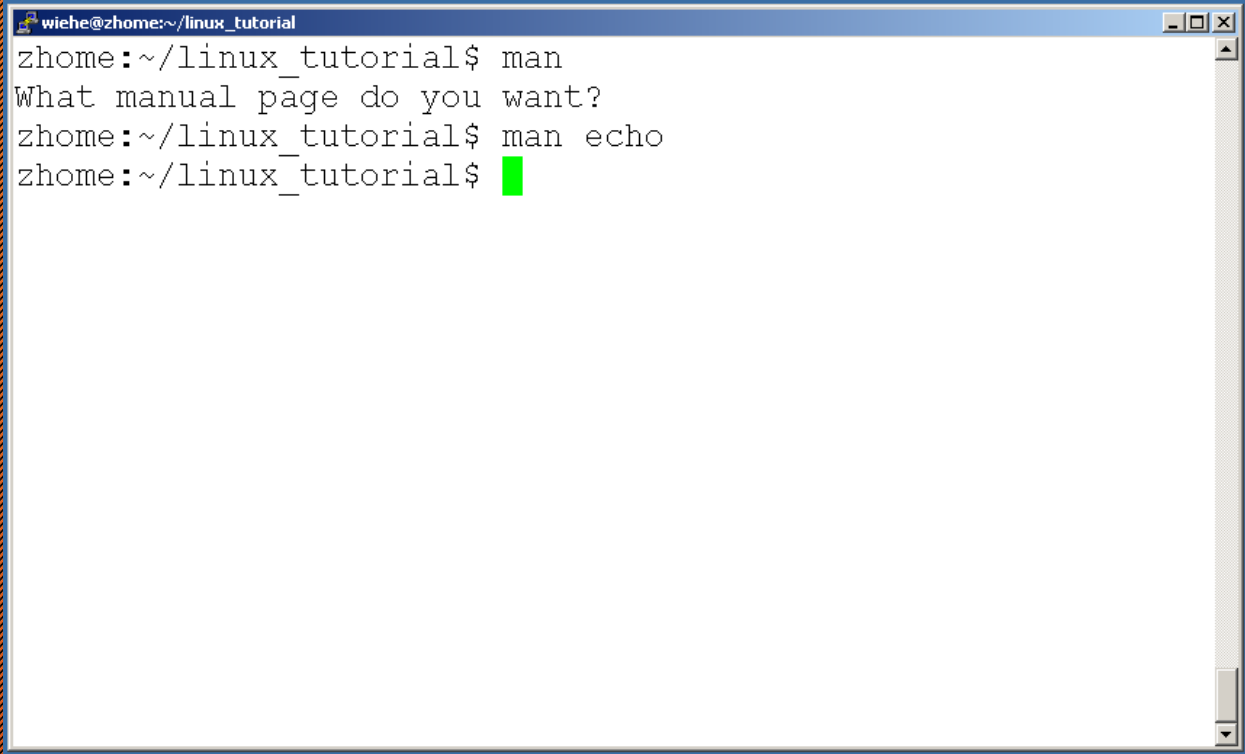

What exactly is a “shell”?

- 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!

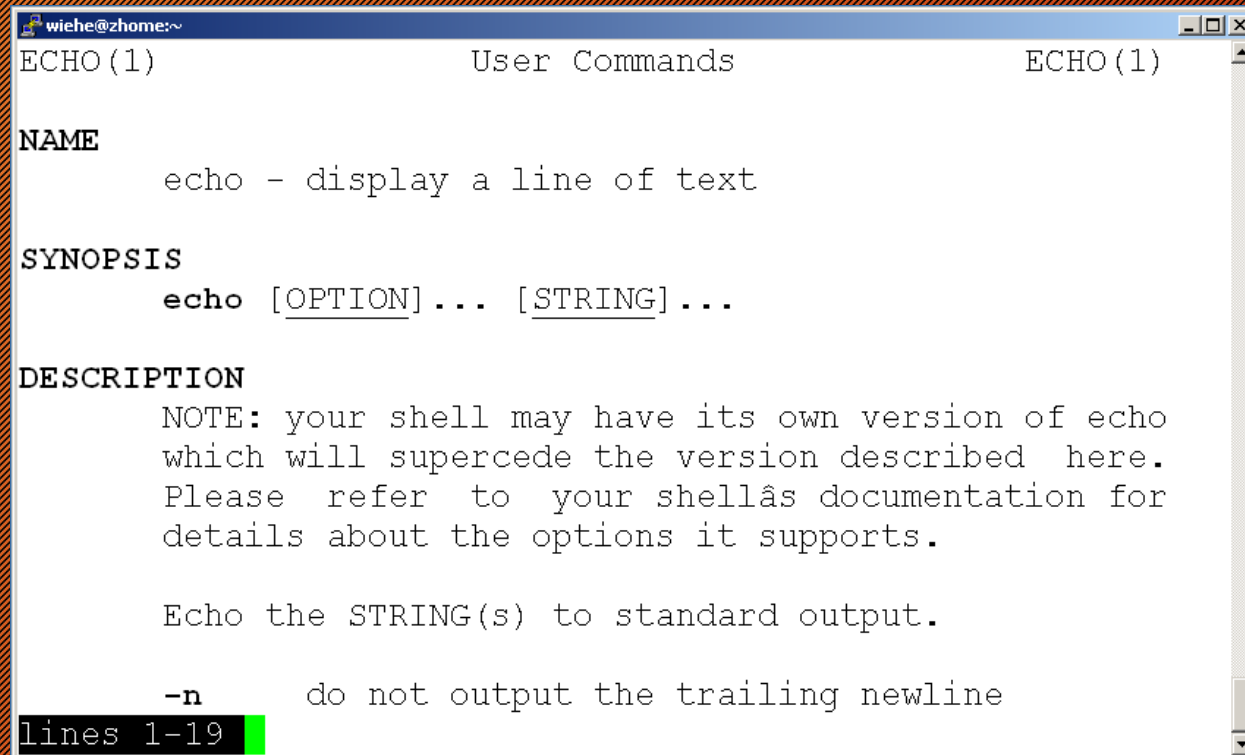


A terminal window titled 'wiehe@zhome:~/linux_tutorial' with standard window controls. The terminal shows the following sequence of commands and output:

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

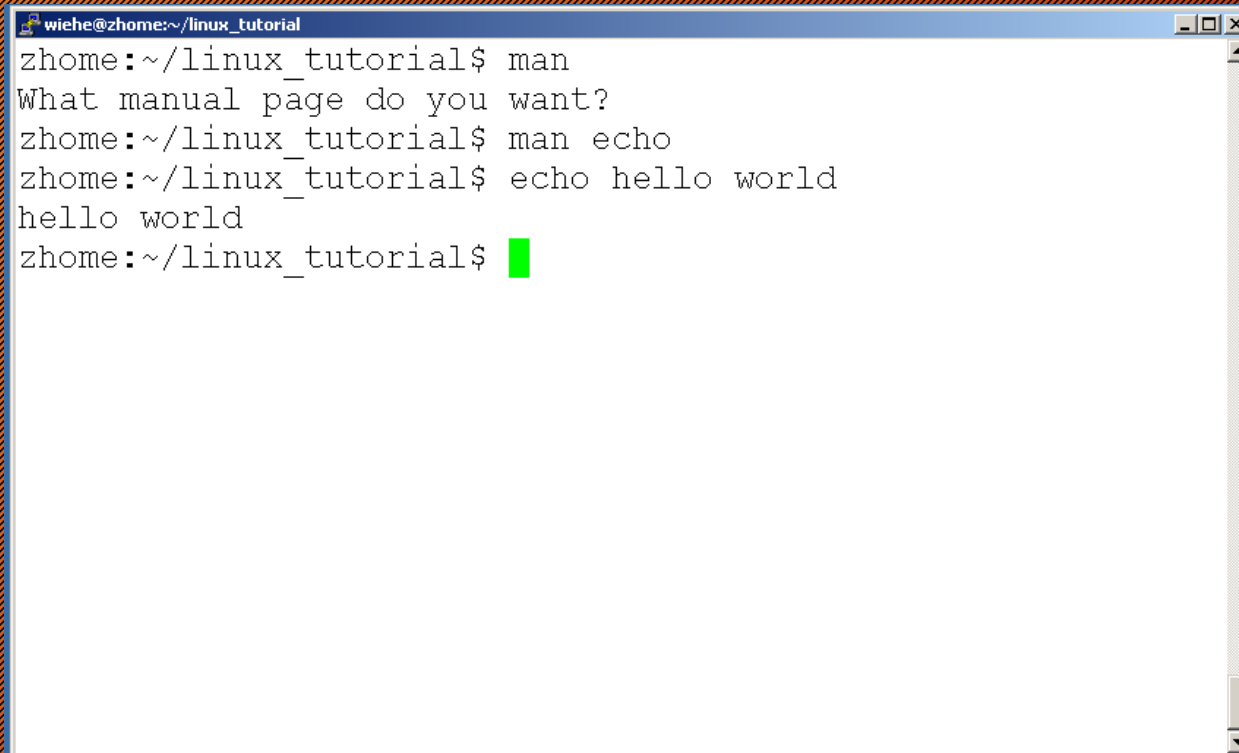
The prompt is green, and the cursor is at the end of the last line.

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
```

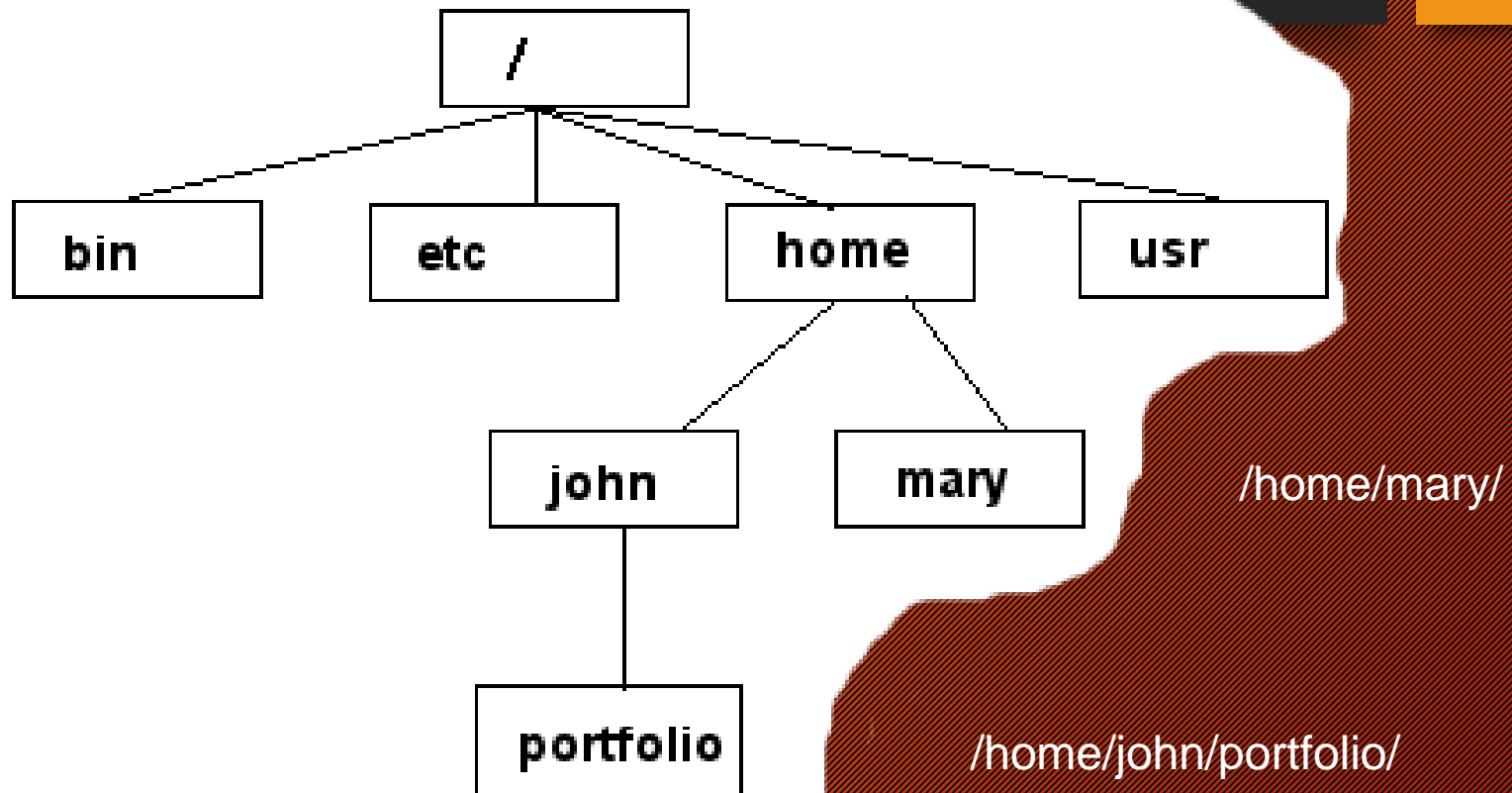
Help!



```
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

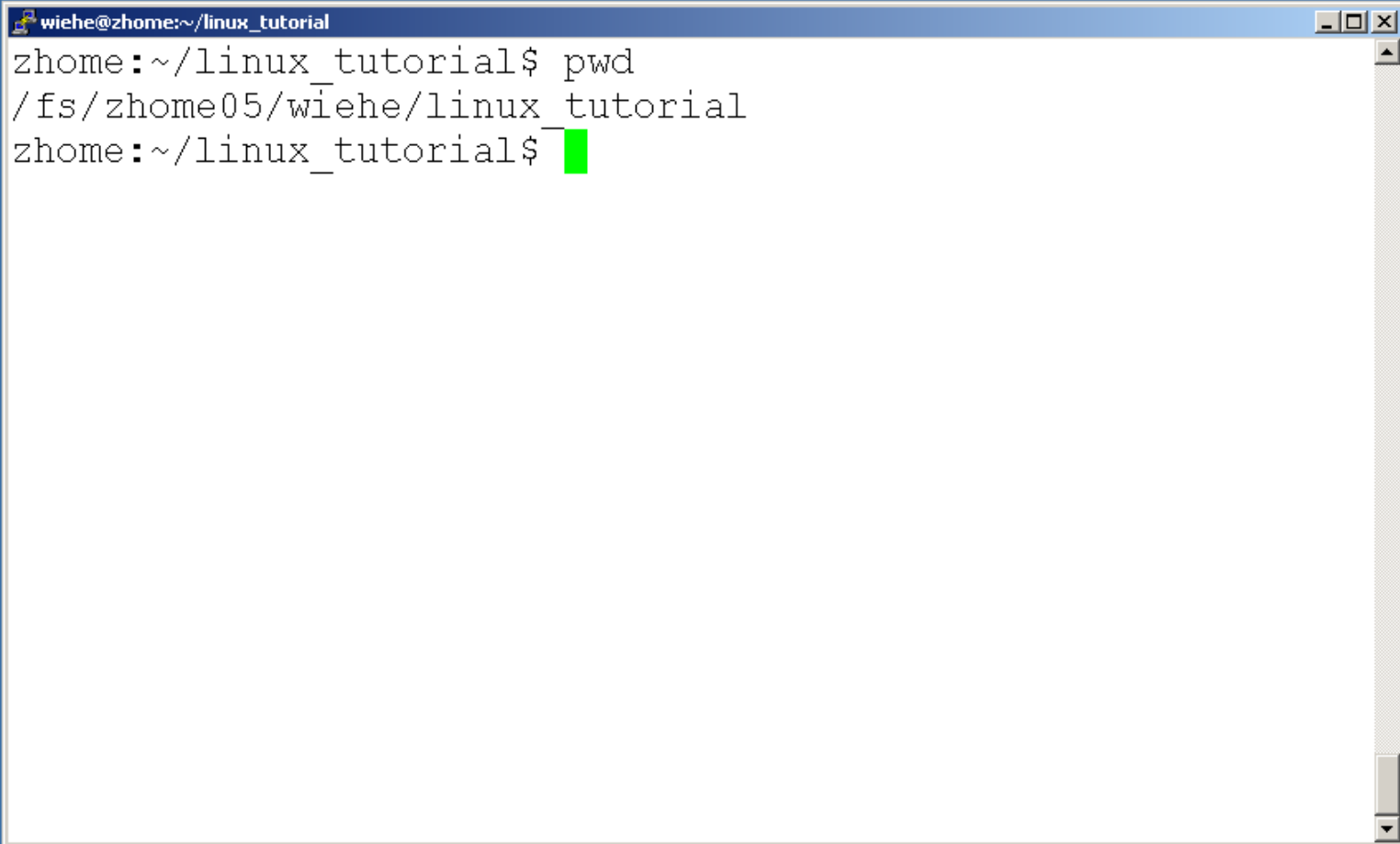
NOTE: Unix file names
are **CASE SENSITIVE!**



The Path

Command: pwd

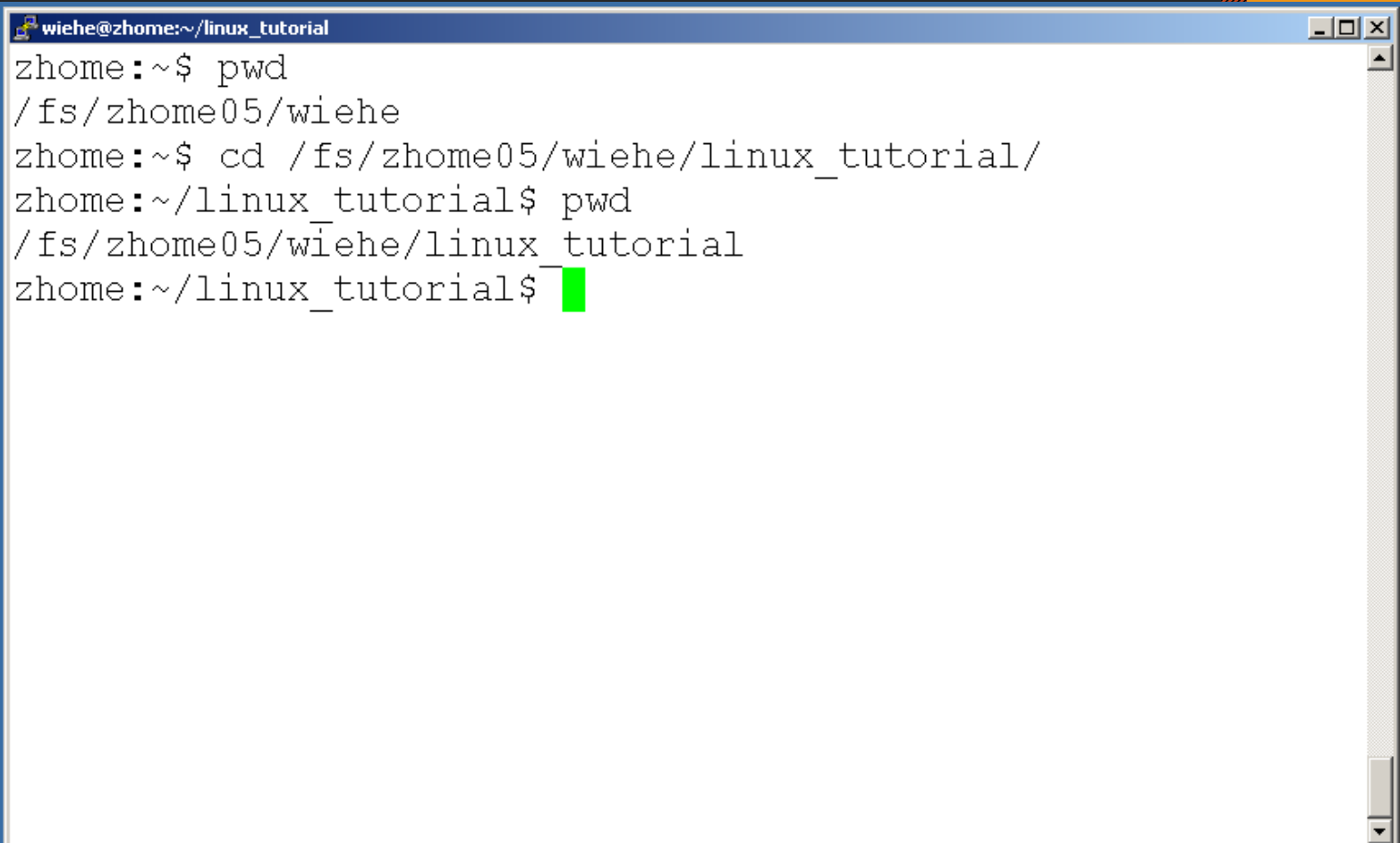
- To find your current path use “pwd”

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 at the 'zhome' prompt in the directory '~/linux_tutorial' typing the command 'pwd'. The output is '/fs/zhome05/wiehe/linux_tutorial'. Below the output, the prompt 'zhome:~/linux_tutorial\$' is shown with a green cursor.

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


Command: cd

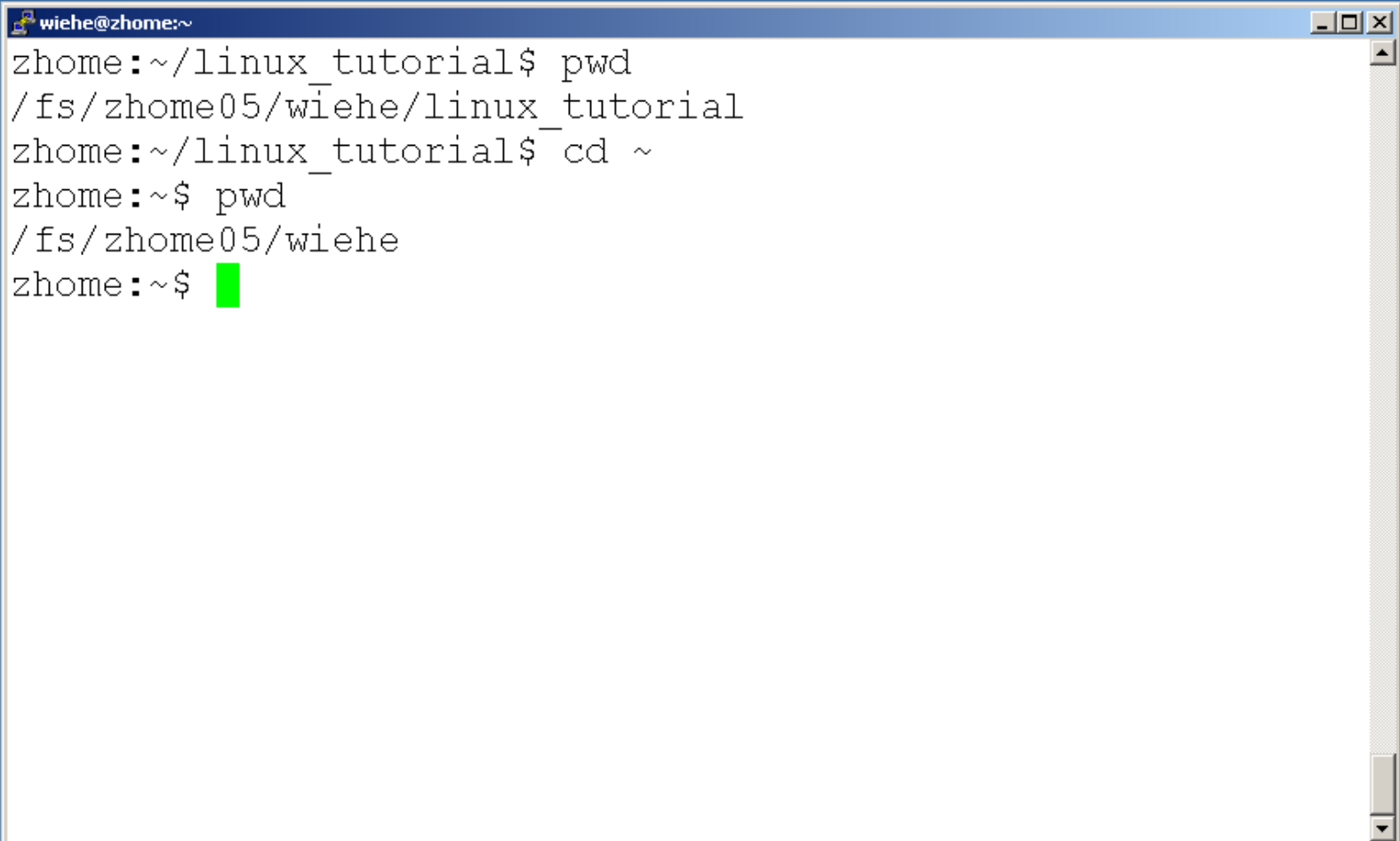
- To change to a specific directory use “cd”

A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing a sequence of commands and their outputs. The user starts in the home directory, runs 'pwd' to confirm the path, then uses 'cd' to move into the 'linux_tutorial' subdirectory. A second 'pwd' command confirms the new directory.

```
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

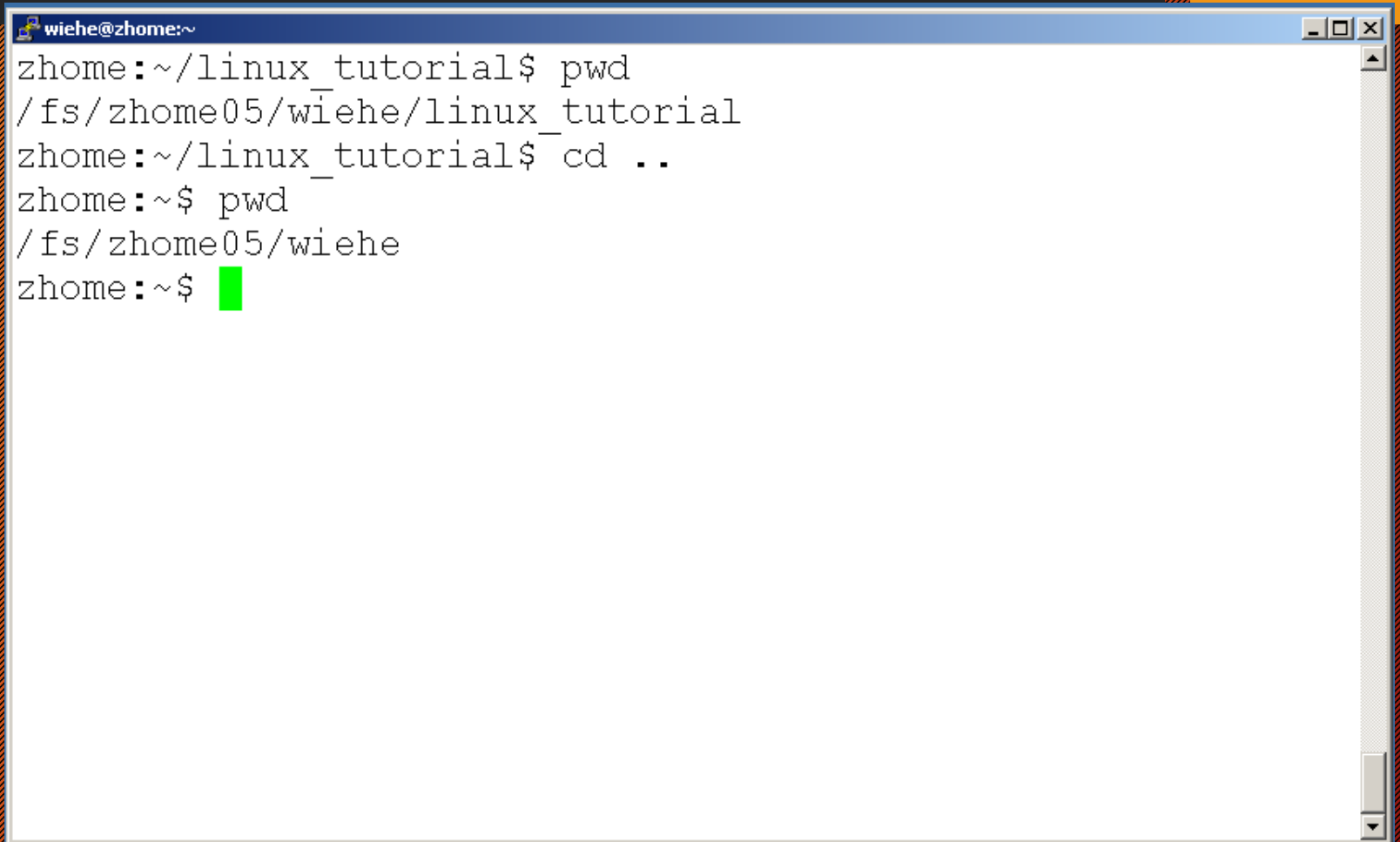
- “~” is the location of your home directory



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

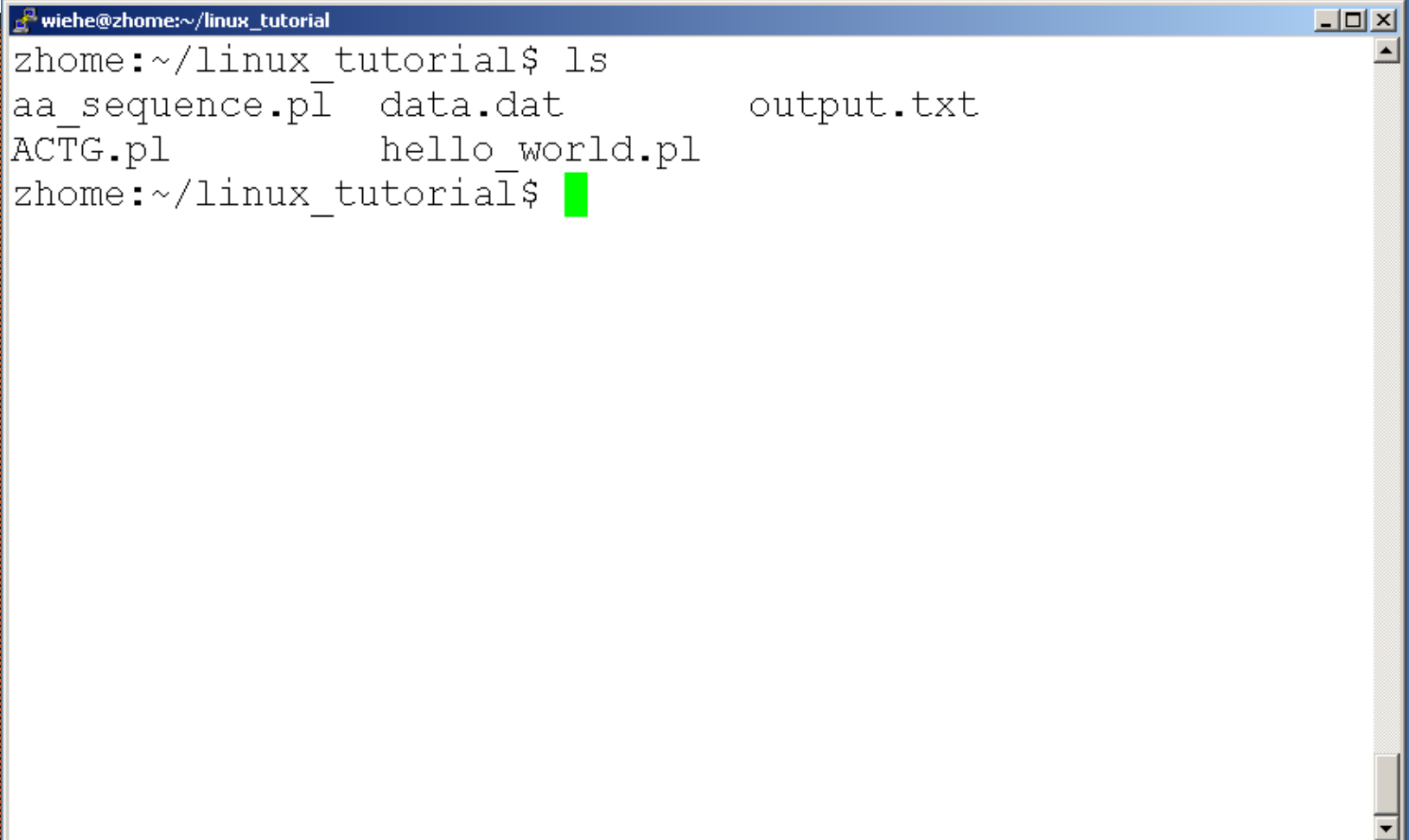

Command: cd

`..` 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

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

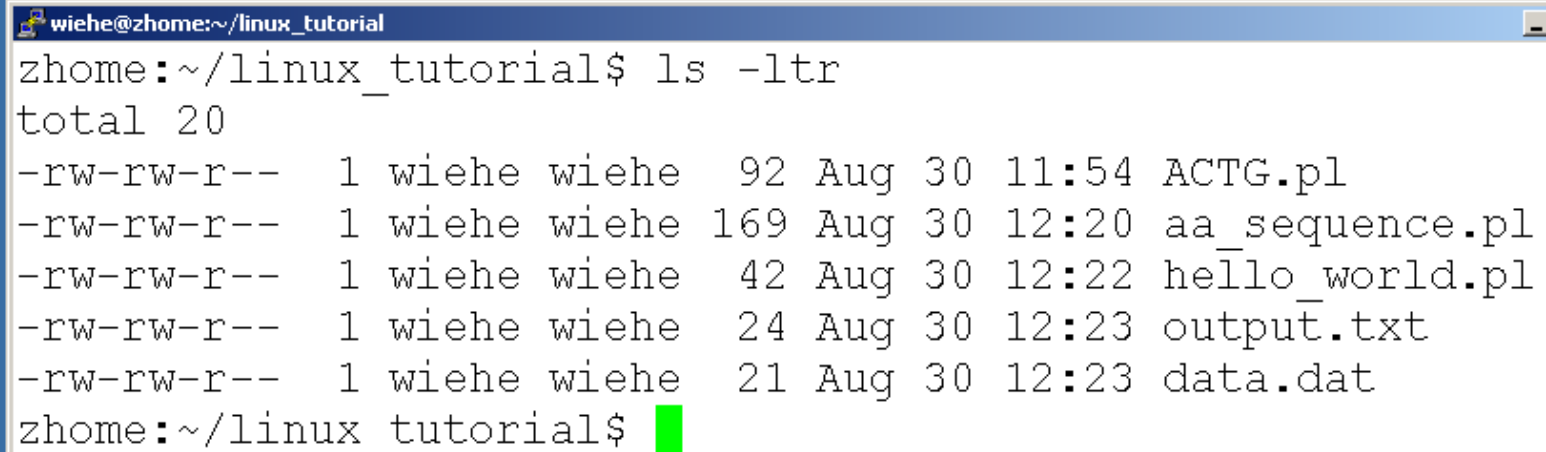
```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$ ls
aa_sequence.pl  data.dat          output.txt
ACTG.pl        hello_world.pl
zhome:~/linux_tutorial$
```


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

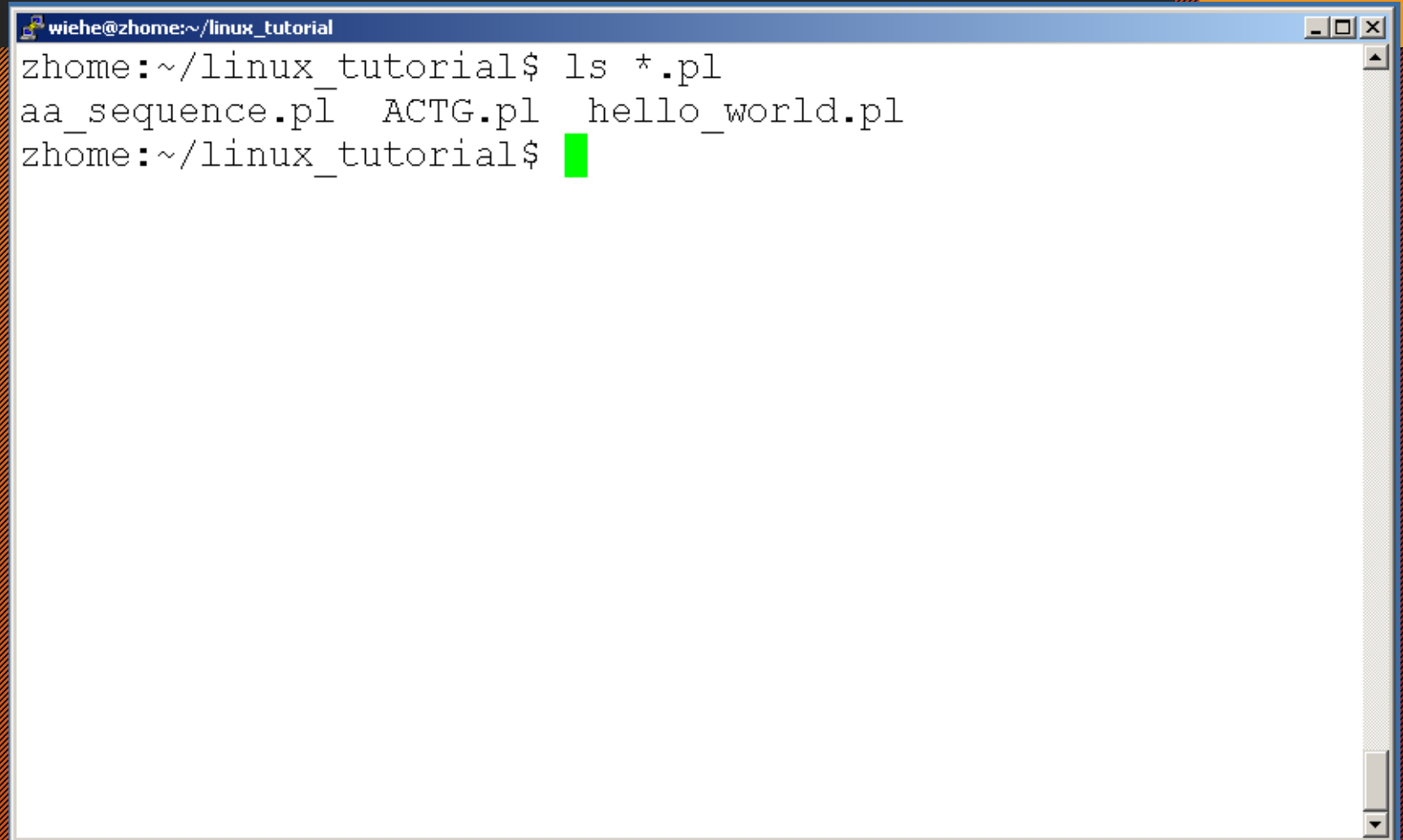


A terminal window titled 'wiehe@zhome:~/linux_tutorial' displays the output of the 'ls -ltr' command. The output lists five files in reverse chronological order, showing permissions, owner, group, size, date, time, and filename. The files are ACTG.pl, aa_sequence.pl, hello_world.pl, output.txt, and data.dat. 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: *

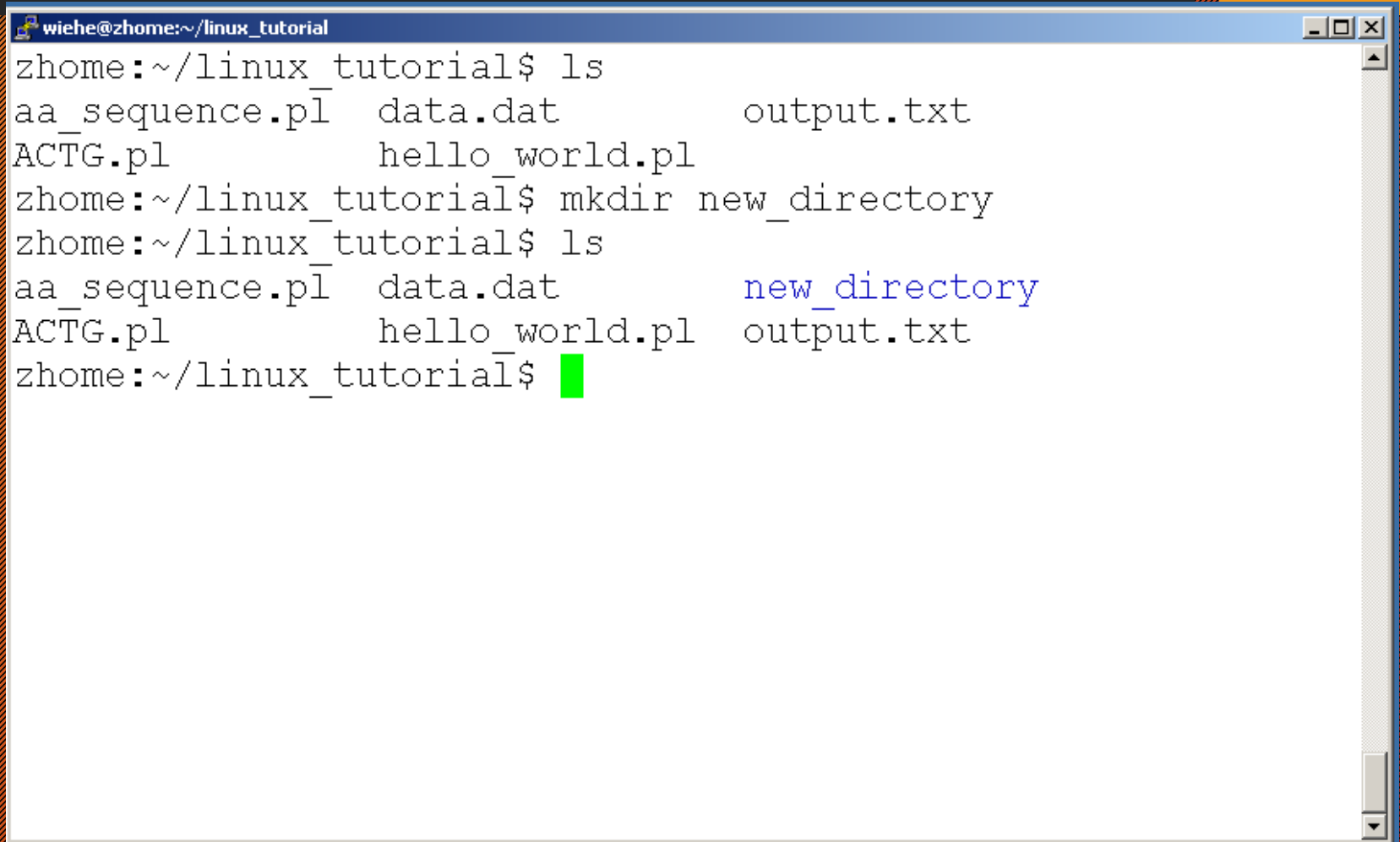
- “*” can be used as a wildcard in unix/linux

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The window has standard window controls (minimize, maximize, close) on the right. The terminal content shows a command 'ls *.pl' being executed, which lists three files: 'aa_sequence.pl', 'ACTG.pl', and 'hello_world.pl'. The prompt 'zhome:~/linux_tutorial\$' is shown at the end of the output line, followed by a green cursor.

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

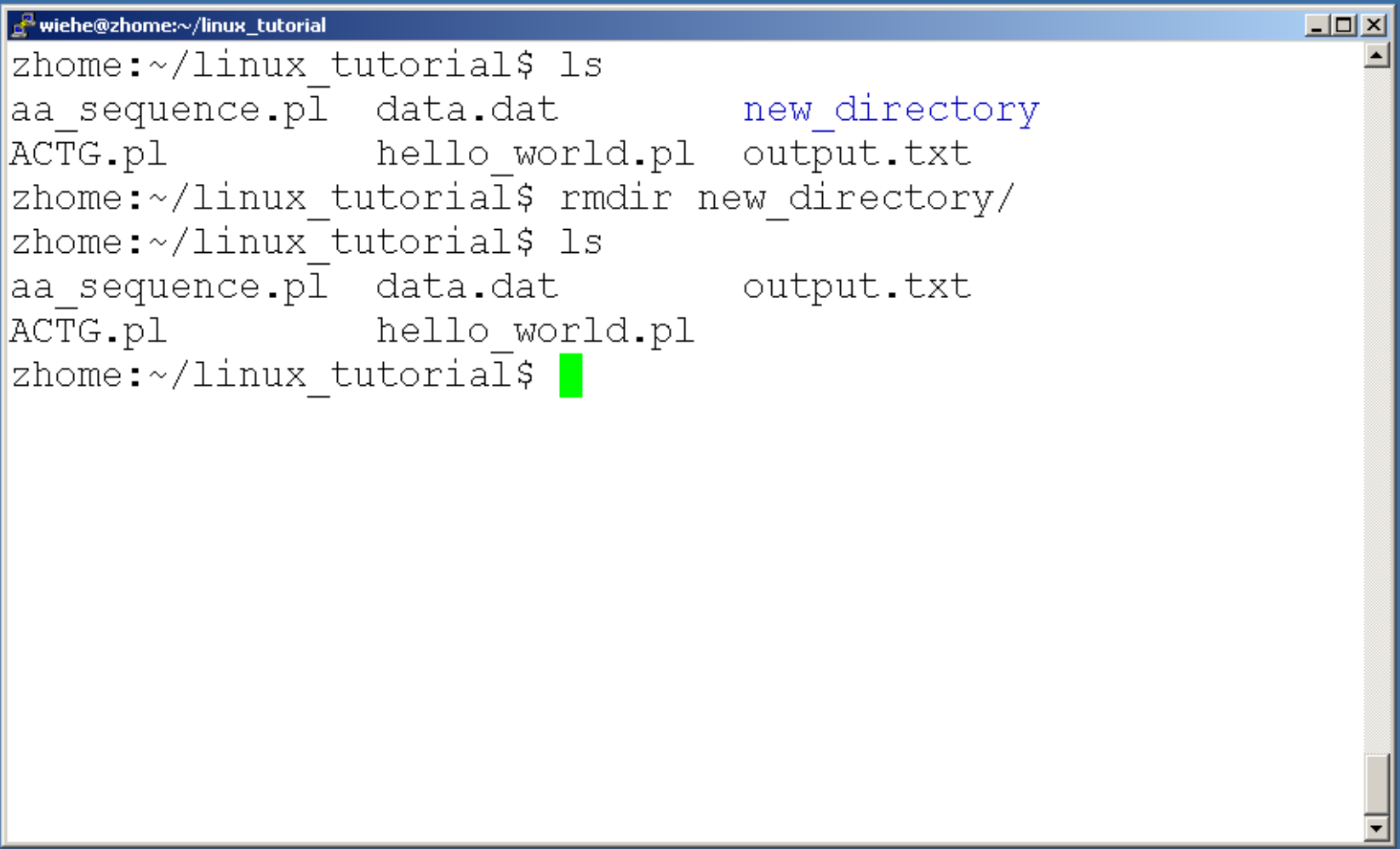
Command: mkdir

- 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' lists 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. A green cursor is at the end of the last prompt.

```
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”

A terminal window titled 'wiehe@zhome:~/linux_tutorial' with standard window controls. It shows a sequence of commands and their outputs. First, 'ls' lists files including 'new_directory'. Then, 'rmdir new_directory/' is executed. Finally, 'ls' is run again, showing that 'new_directory' has been removed. A green cursor is at the end of the last command line.

```
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$
```

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

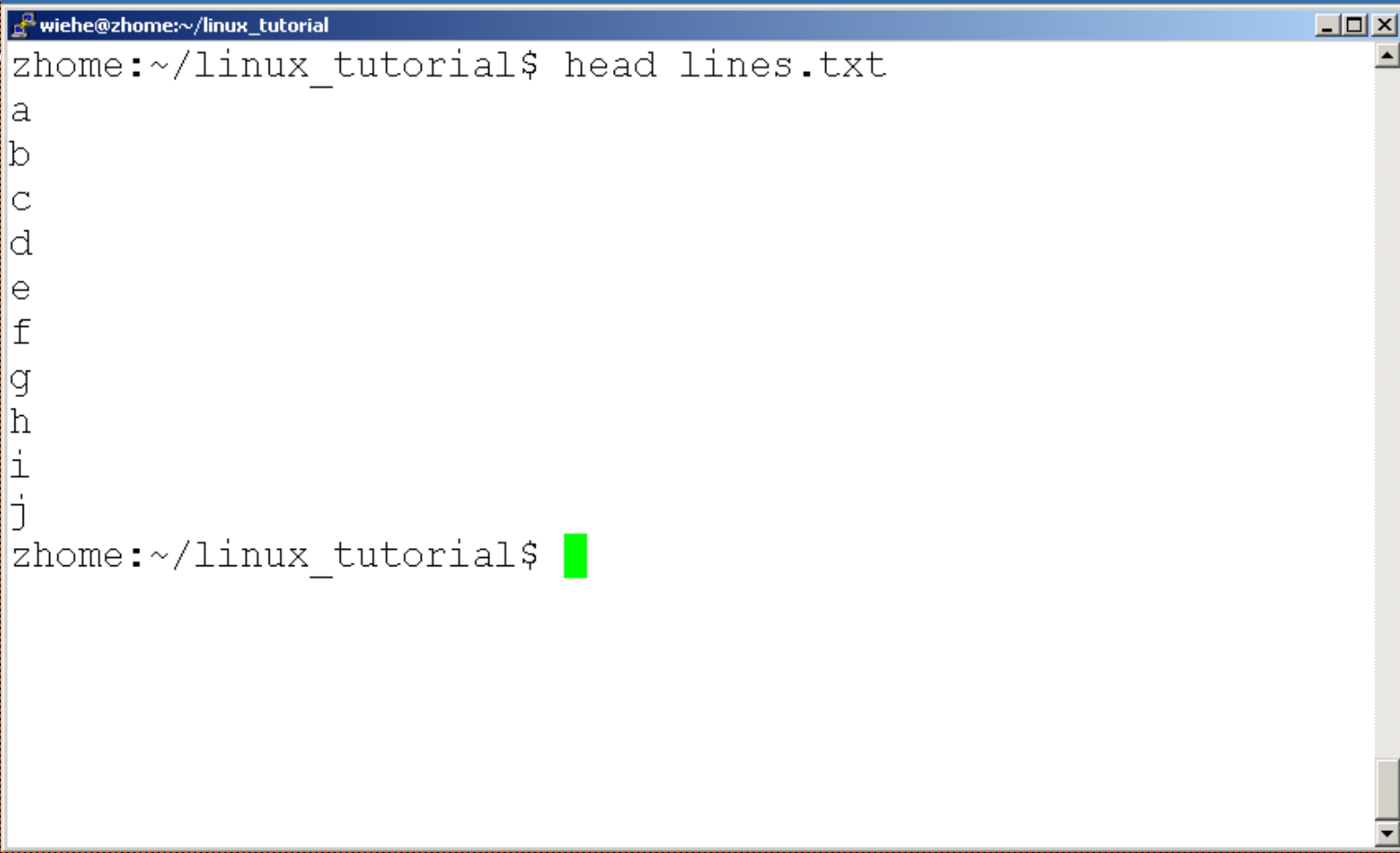
Command: less

- “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

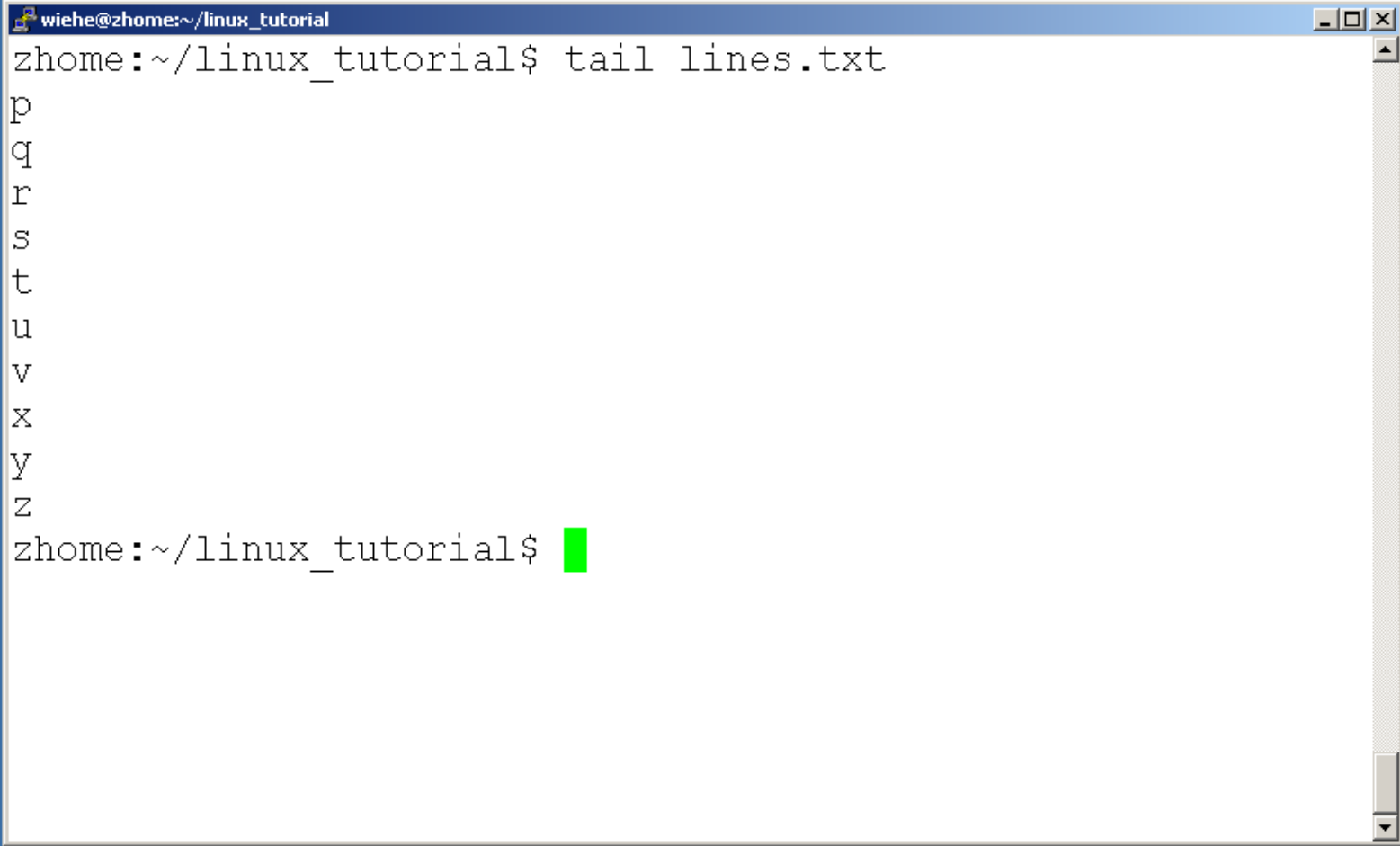
Command: head

- A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing the command 'head lines.txt' being executed. The output displays the first ten lines of the file 'lines.txt', which are the lowercase letters 'a' through 'j'. The prompt 'zhome:~/linux_tutorial\$' is shown at the bottom with a green cursor.

```
wiehe@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

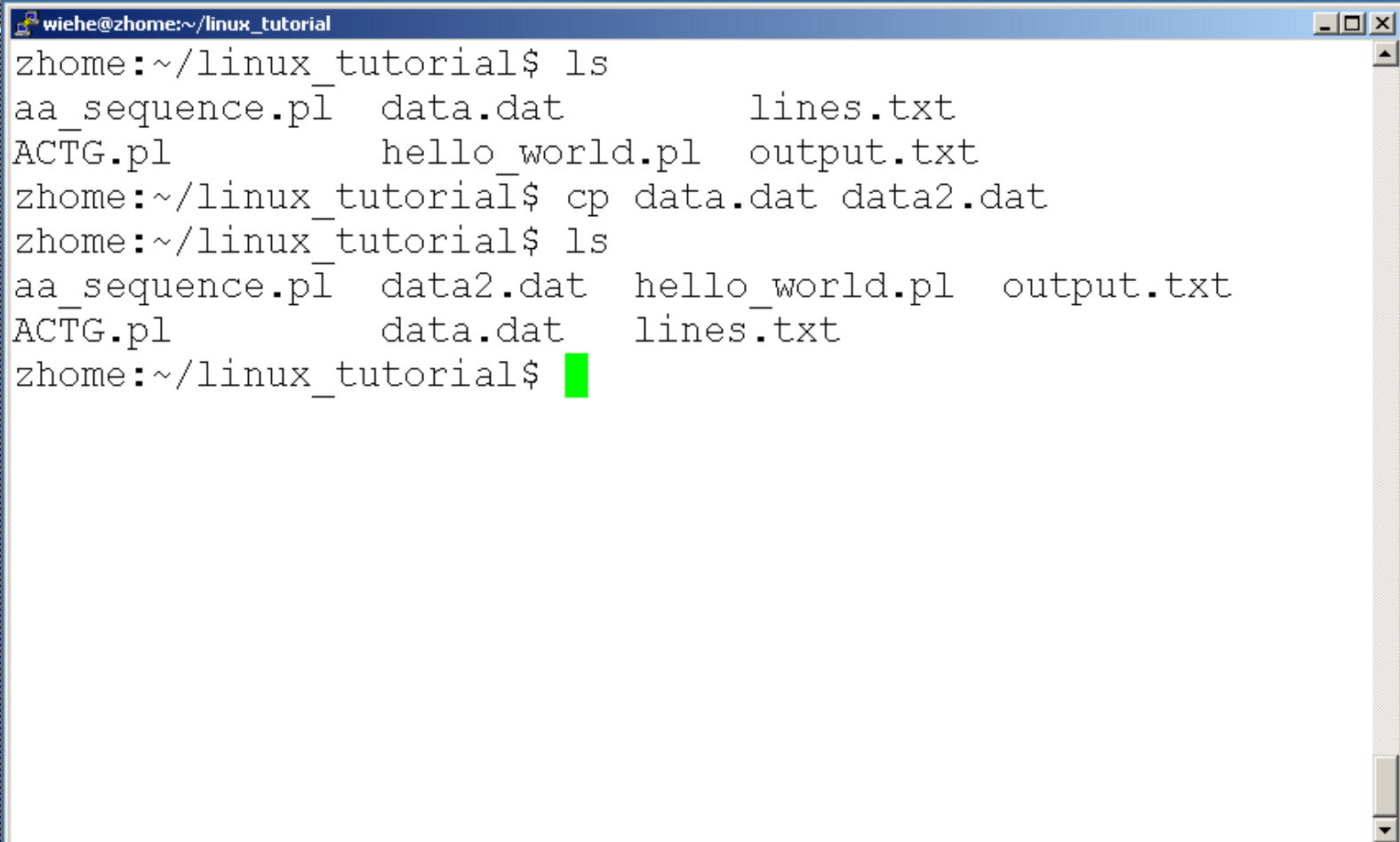
A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The window shows the command 'tail lines.txt' being executed. The output consists of the letters 'p', 'q', 'r', 's', 't', 'u', 'v', 'x', 'y', and 'z' on separate lines. The prompt 'zhome:~/linux_tutorial\$' is 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`

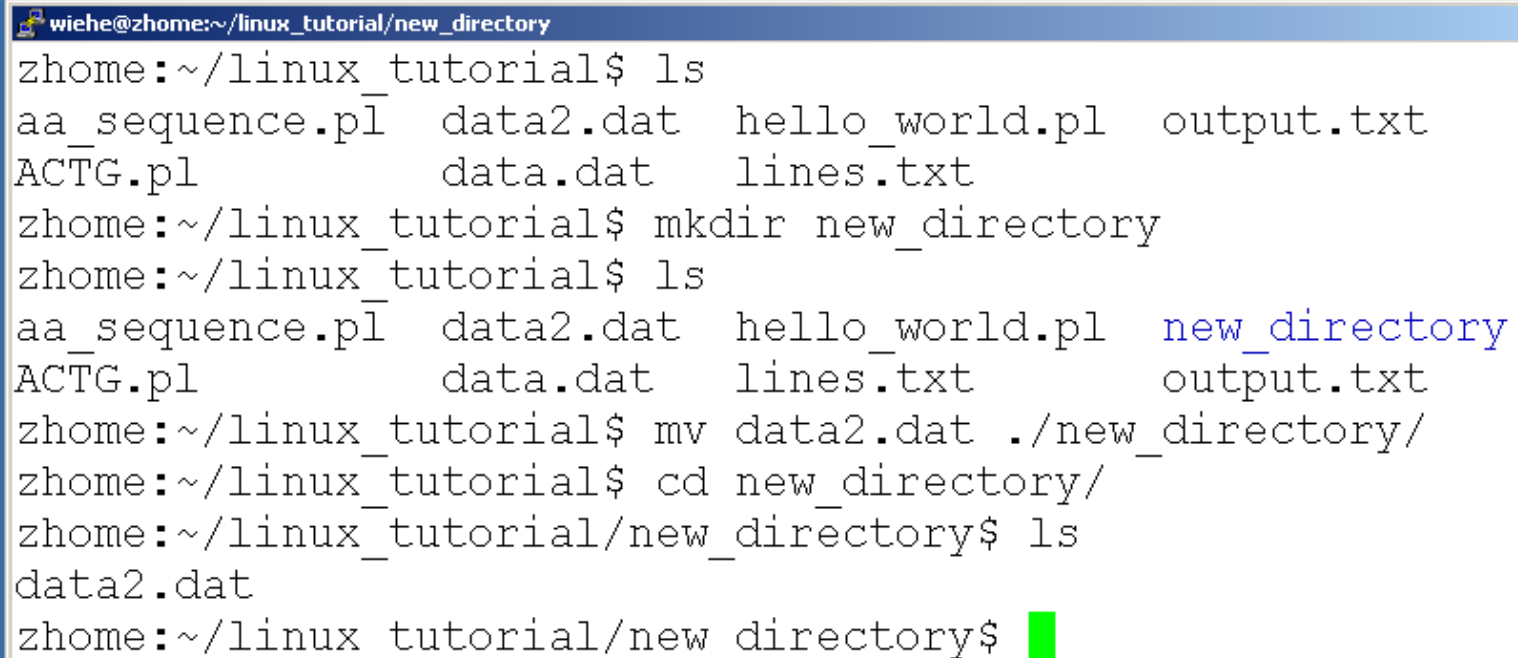
Command: cp

- A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing the execution of the 'cp' command. The user first lists the directory contents, then copies 'data.dat' to 'data2.dat', and lists the directory again to confirm the copy.

```
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

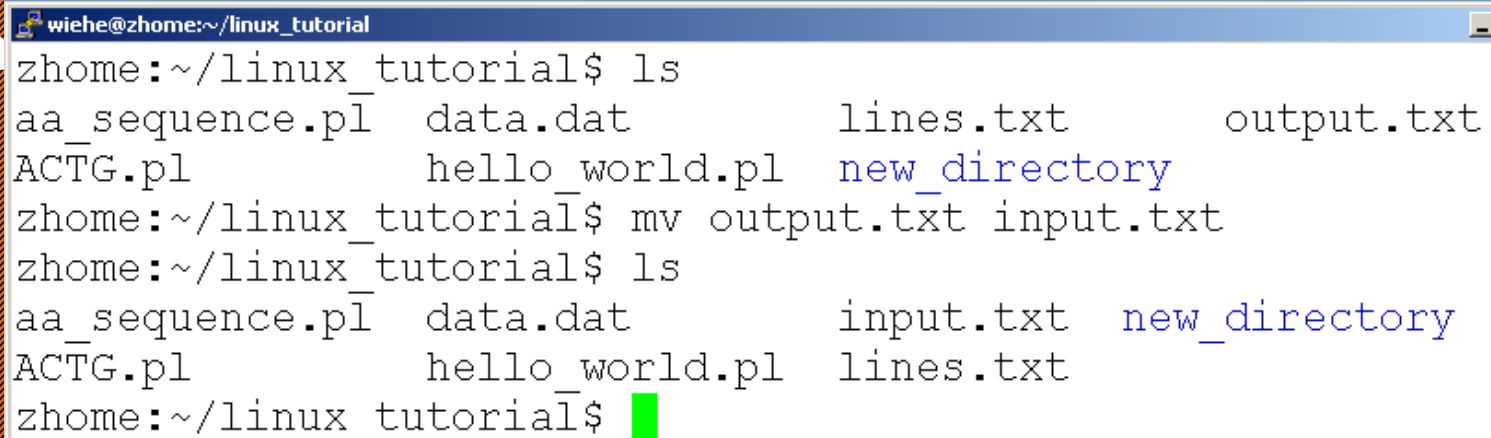
- To move a file to a different location use

A terminal window titled 'wiehe@zhome:~/linux_tutorial/new_directory' showing a sequence of commands and their outputs. The commands include 'ls', 'mkdir new_directory', 'ls', 'mv data2.dat ./new_directory/', 'cd new_directory/', and another 'ls'. The output shows the directory structure and the successful movement of 'data2.dat' into the 'new_directory' subdirectory. A green cursor is visible at the end of the final command line.

```
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

- n

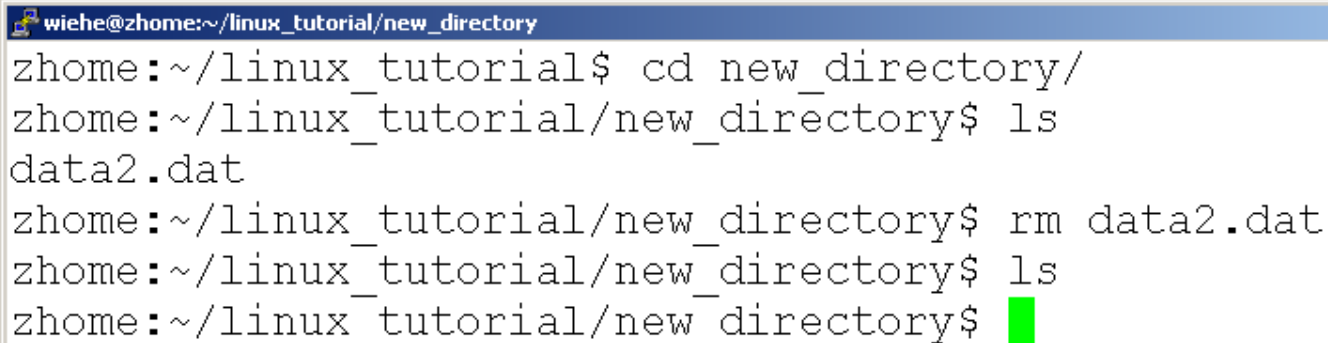


A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing the execution of the 'mv' command. The window has a blue title bar and standard window controls. The terminal output shows the initial directory listing, the command to move 'output.txt' to 'input.txt', and the resulting directory listing where 'output.txt' is replaced by 'input.txt'.

```
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 window shows a series of commands and their outputs. The user navigates to the "new_directory" folder, lists its contents (showing "data2.dat"), and then uses the "rm" command to remove "data2.dat". A subsequent "ls" command confirms the file has been removed. The prompt ends with a green cursor.

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

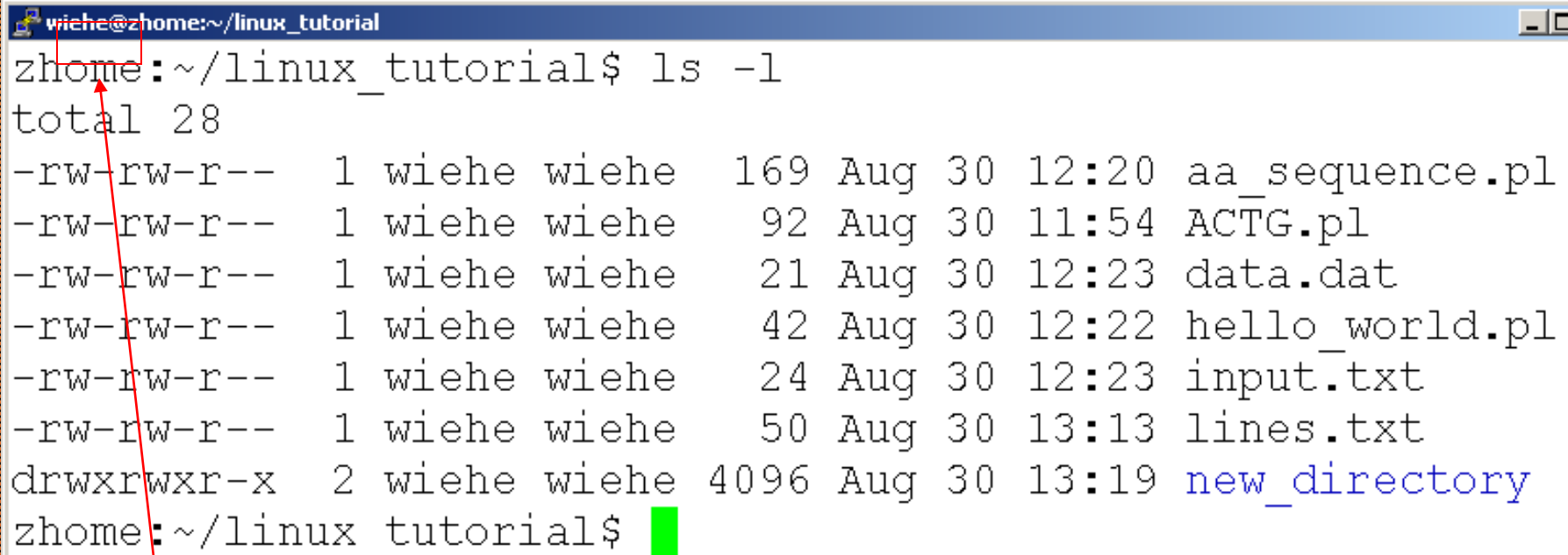
File permissions

- 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

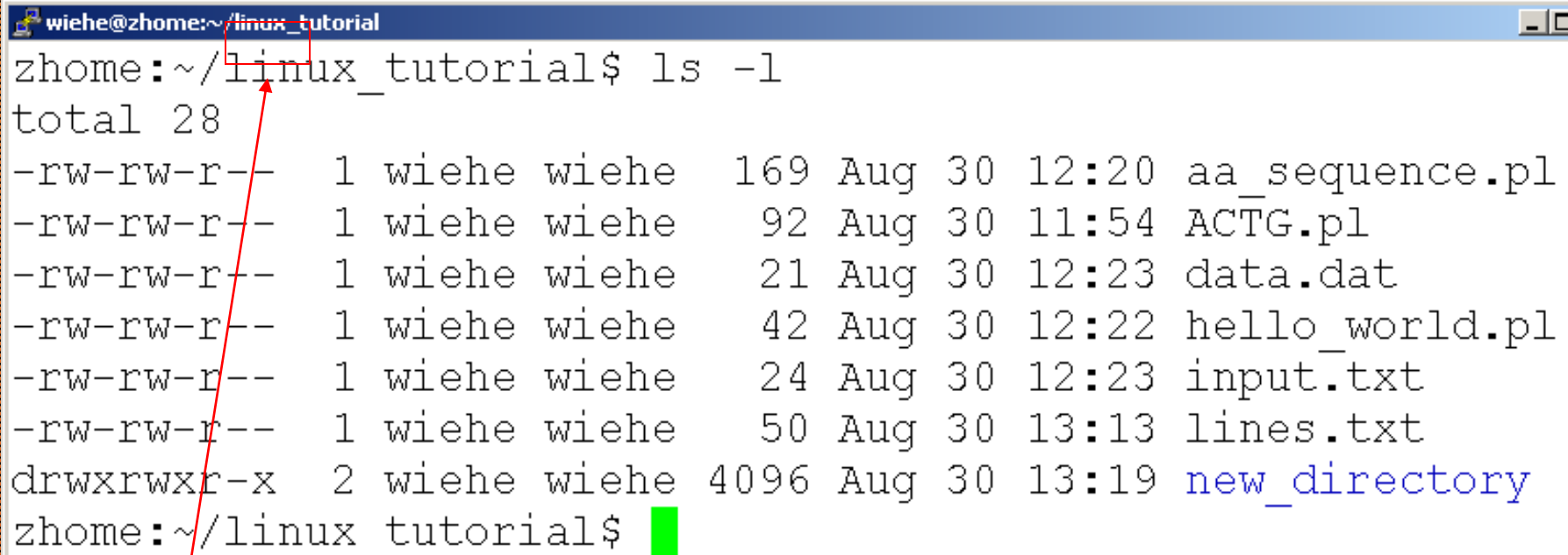
A terminal window titled 'wiehe@zhome:~/linux_tutorial' displays the output of the 'ls -l' command. The output lists several files with their permissions, owner, group, size, date, and filename. A red arrow points from the first file's permissions to the 'zhome' part of the prompt. A green cursor is at the end of the last prompt line.

```
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$
```


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$
```

File Permissions

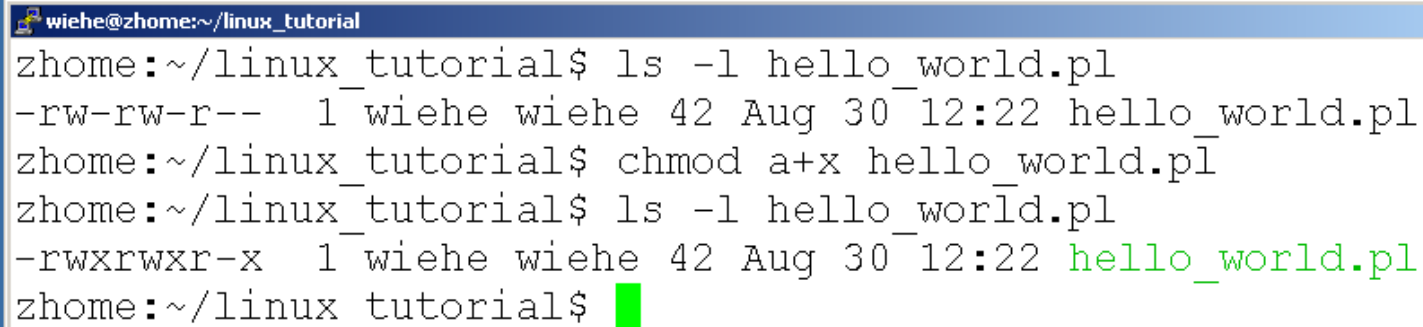


A terminal window titled 'wiehe@zhome:~/linux_tutorial' displays the output of the command 'ls -l'. The output lists several files with their permissions, owner, group, size, date, and name. A red arrow points from the first file's permissions to the 'ls' command in the prompt. The prompt ends with a green cursor.

```
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$
```


Command: chmod

- If you own the file, you can change it's permissions with "chmod"
 - Syntax: `chmod [User/group/others/all]+[permission]`

A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing the process of changing file permissions. The user runs 'ls -l hello_world.pl' to see the current permissions '-rw-rw-r--'. Then, they run 'chmod a+x hello_world.pl' to add execute permissions. Finally, they run 'ls -l hello_world.pl' again, showing the updated permissions '-rwxrwxr-x'. The filename 'hello_world.pl' is highlighted in green in the second and third 'ls' outputs.

```
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$
```

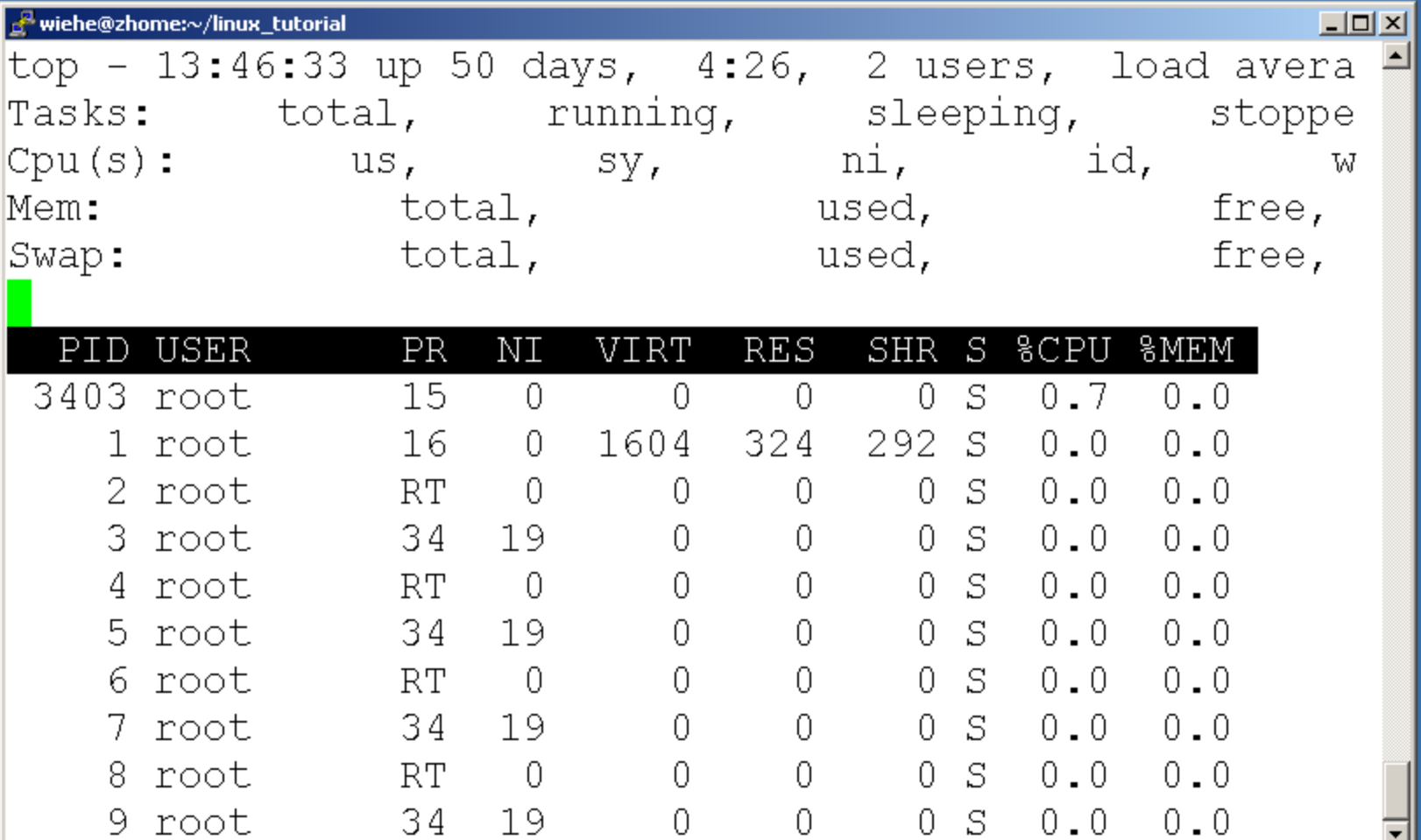
Command: ps

- `T` 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\$

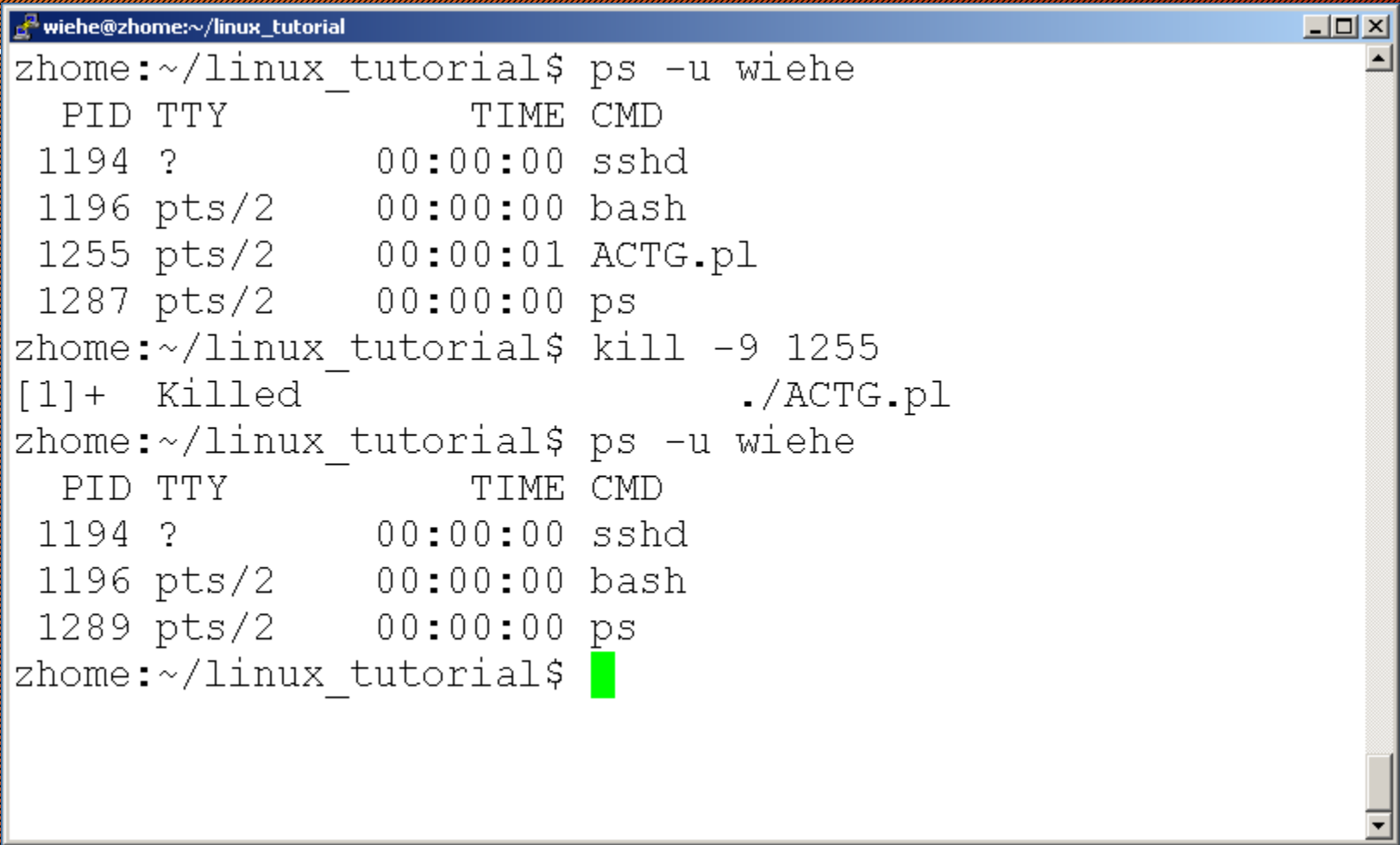
Command: top

- 

```
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

- A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing the execution of the 'kill' command. The user first runs 'ps -u wiehe' to list processes. Then, they run 'kill -9 1255' to kill a process with PID 1255. The terminal shows '[1]+ Killed ./ACTG.pl'. Finally, they run 'ps -u wiehe' again to show the updated process list.

```
wiehe@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
wiehe@zhome:~/linux_tutorial$ kill -9 1255
[1]+  Killed                  ./ACTG.pl
wiehe@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
wiehe@zhome:~/linux_tutorial$
```


Input/Output Redirection ("piping")

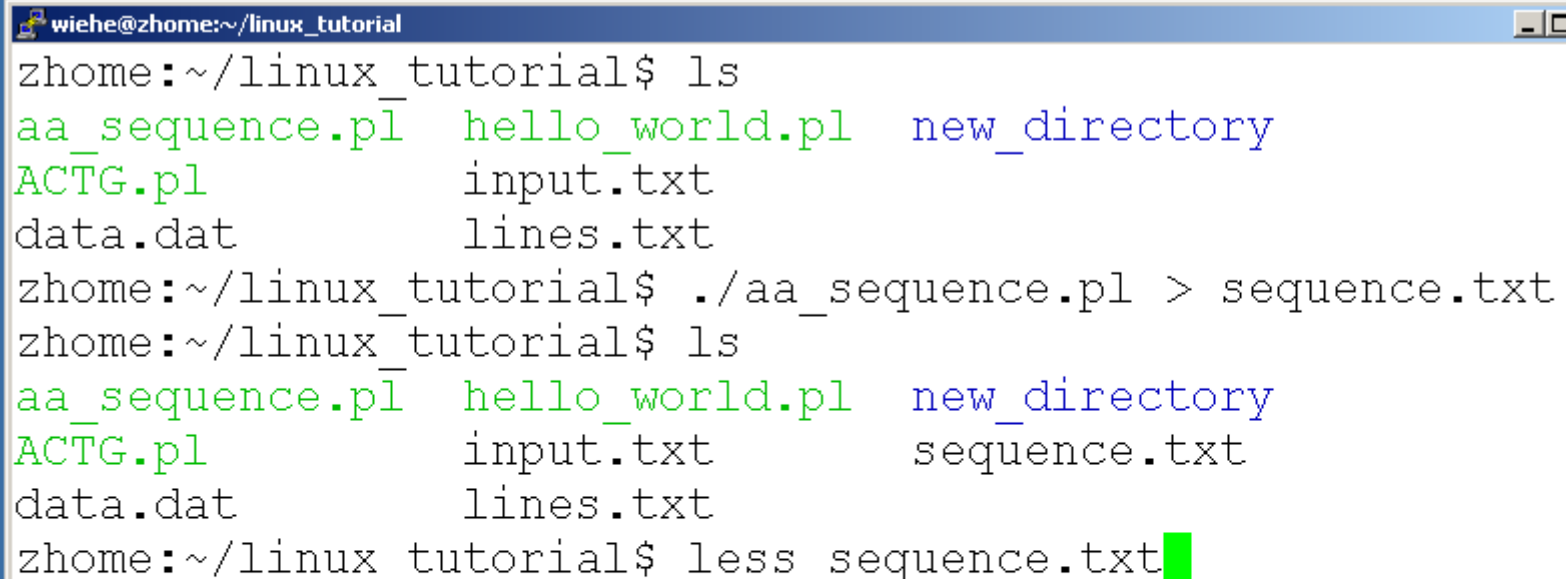
- Programs can output to other programs
- Called "piping"
- "program_a | program_b"
 - program_a's output becomes program_b's input
- "program_a > file.txt"
 - program_a's output is written to a file called "file.txt"
- "program_a < input.txt"
 - program_a gets its input from a file called "input.txt"

A few examples of piping

wiehe@zhome:~/linux_tutorial

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


A few examples of piping



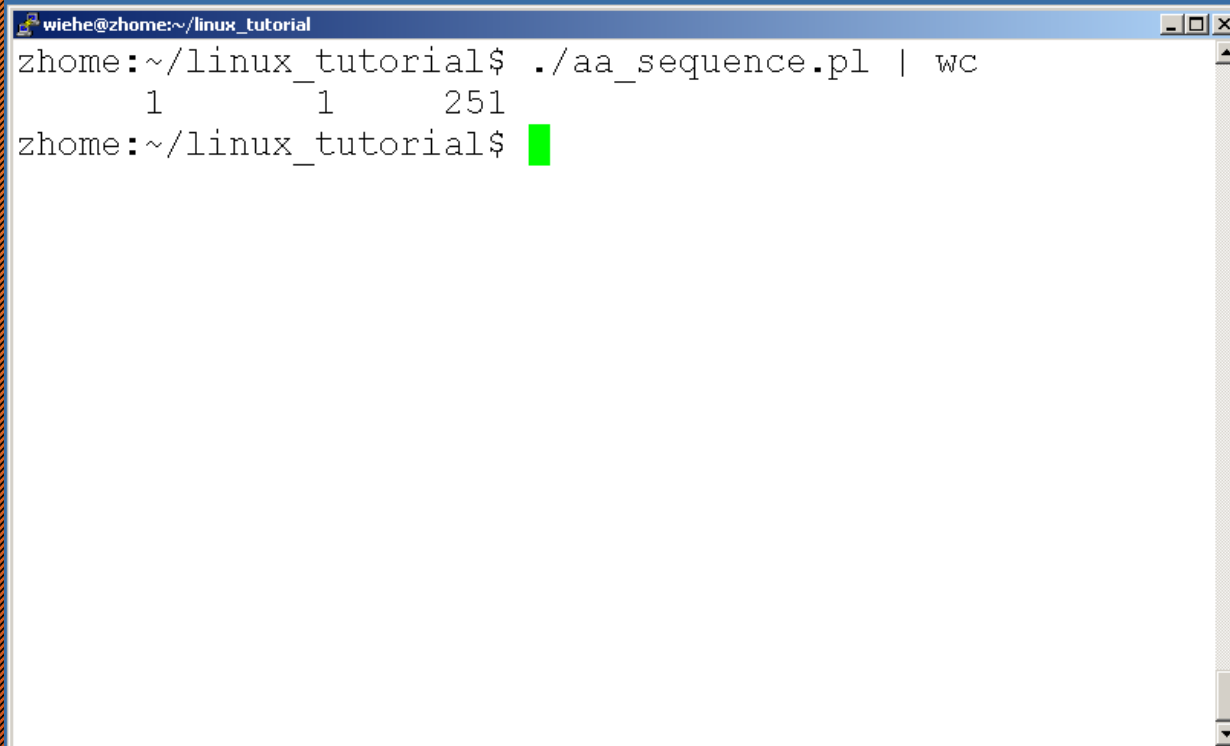
A terminal window titled 'wiehe@zhome:~/linux_tutorial' displays the following commands and output:

```
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

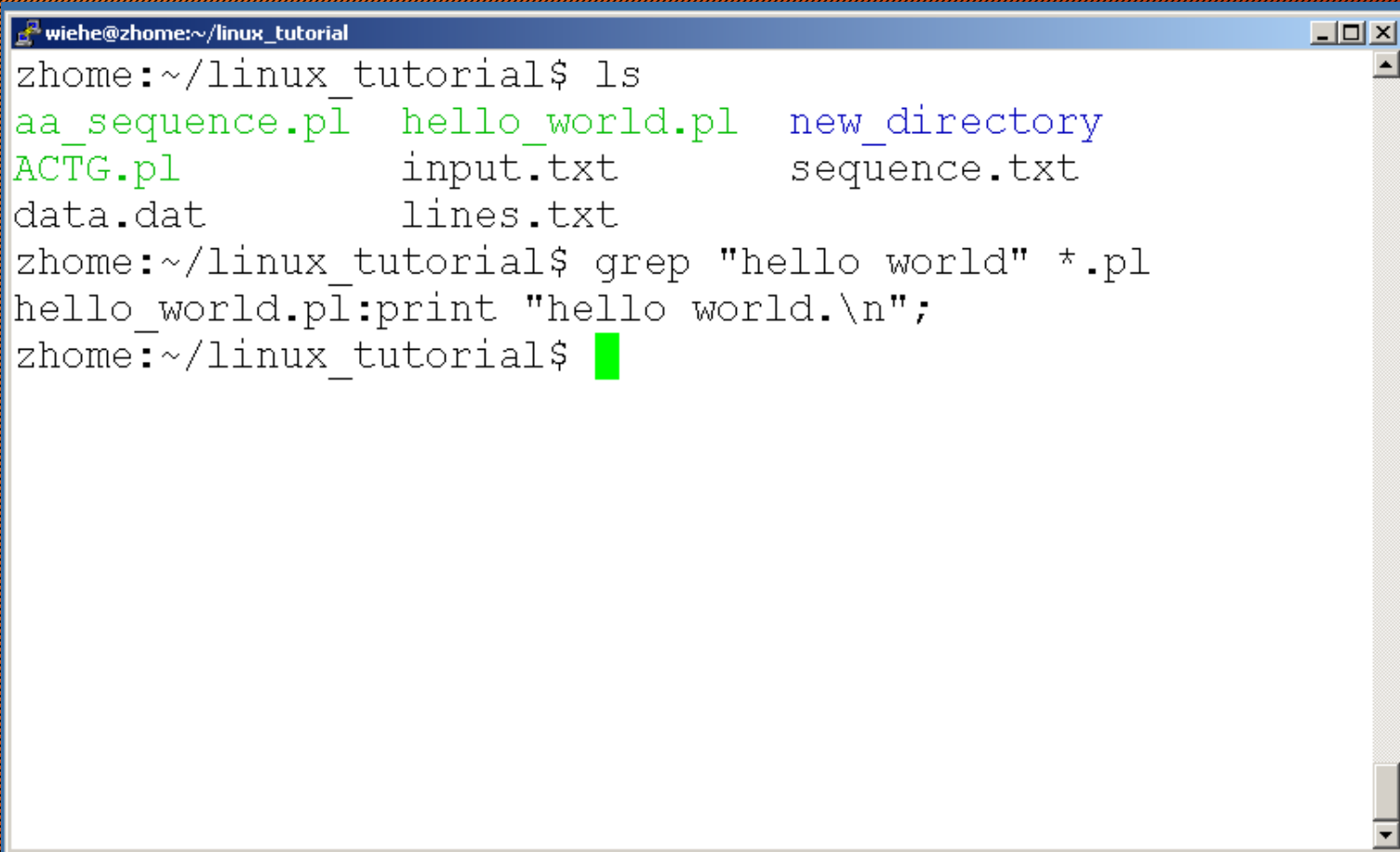
A few examples of piping

A terminal window with a blue title bar containing the text 'wiehe@zhome:~/linux_tutorial'. The window shows a command being executed and its output. The command is './aa_sequence.pl | wc'. The output is '1 1 251'. The prompt 'zhome:~/linux_tutorial\$' is shown twice, once before the command and once after the output.

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

Command: grep

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

A terminal window titled 'wiehe@zhome:~/linux_tutorial' showing a series of commands and their outputs. The user first runs 'ls' to list files in the current directory. Then, they run 'grep "hello world" *.pl' to search for the string 'hello world' in all files with a '.pl' extension. The output shows that 'hello_world.pl' contains the string.

```
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