UNIX SYSTEMS PROGRAMMING

By

Dr. Trilok Nath Pandey

Dept. of C.S.E

S ‘O’ A, Deemed to be University

ITER, Bhubaneswar.

Today’s Agenda

IO Redirection

- Filters
- – wc
- – sort
- – head
- – tail
- – grep
- – pipe
- – tee

IO Redirection

- Operating system defines a standard input device and a standard output device
- UNIX defines keyboard to be the standard input device and the monitor to be the standard output device
- If a command is defined to take input from the standard input, it takes input from the keyboard
- If a command gives its output to the standard output, it displays the output to the monitor

IO Redirection

- UNIX allows to temporarily change the standard input and standard output by means of what is called as Indirection & Piping
- – The symbol > means indirection of output (to a file or printer)
- – the symbol < means indirection of input for a command (from a file)

IO Redirection

- Create a file named ***testfile*** with the following content
- ***A person who never made a mistake never tried anything new.***
- Save and close (:wq) the file.
- Execute the following command:
- [SysPgm@labserver ~] $ cat testfile > file2↵
- – The above command declares ***file2*** as the temporary standard output
- – contents of ***testfile*** to be redirected to ***file2***

IO Redirection

- [SysPgm@labserver ~] $ cat testfile > file2↵
- – If the file file2 is not empty, it will be overwritten
- – To avoid this, use >>
- This appends to the old contents of file2
- [SysPgm@labserver ~] $ cat testfile >> file2↵

IO Redirection

- Input redirection
- [SysPgm@labserver ~] $ cat testfile↵
- – Takes input from the file named testfile and displays the result into the standard output
- This command will not take input from standard input
- [SysPgm@labserver ~] $ cat < testfile↵
- – No difference from the previous command
- – ***testfile*** is temporarily the standard input and the command cat gets its input from standard input
- [SysPgm@labserver ~] $ cat < testfile > op↵
- [SysPgm@labserver ~] $ cat testfile > op↵

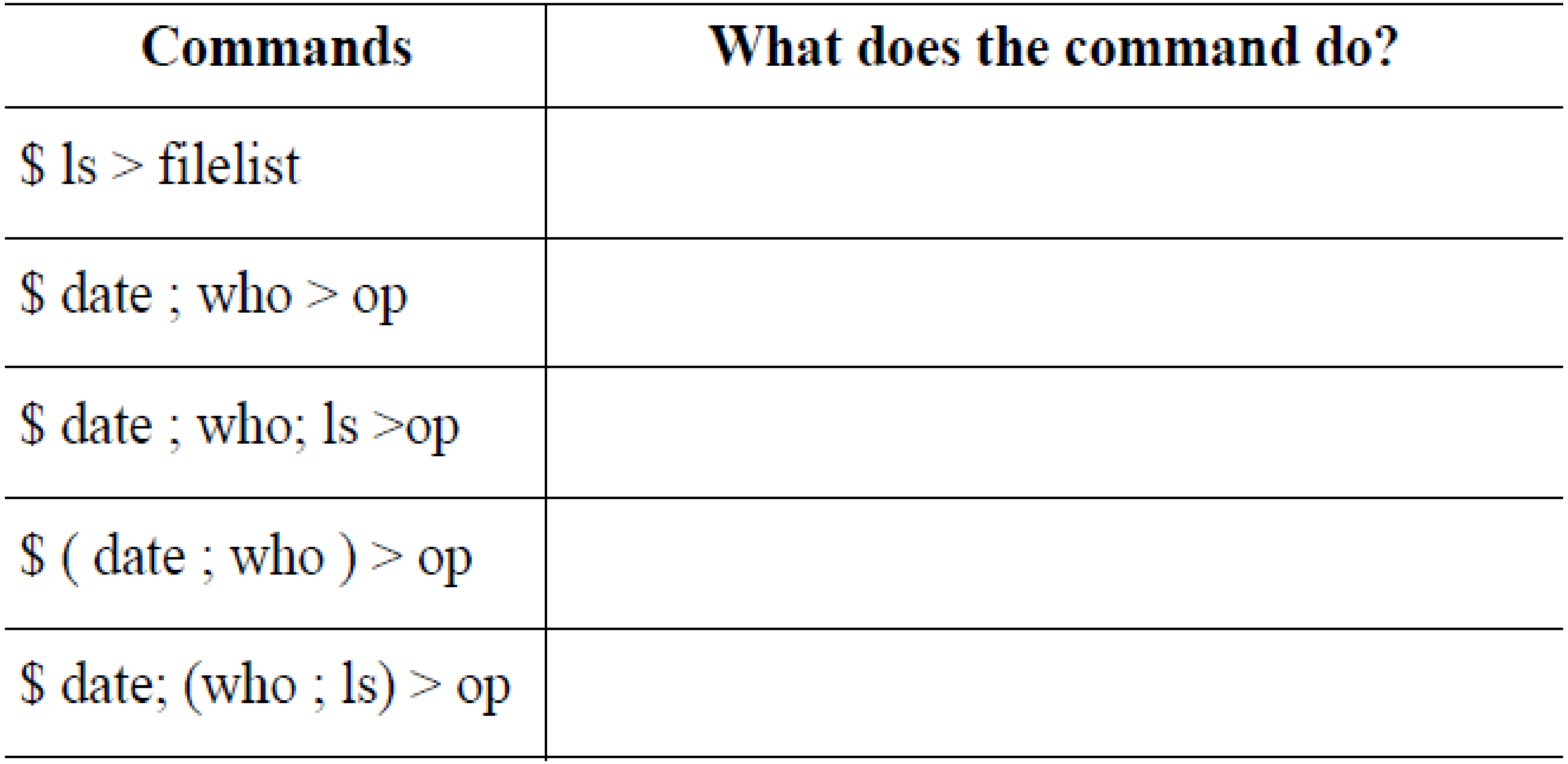
IO Redirection

- Indirect operators

Command	Function
> file	make file as the standard output
< file	make file as the standard input
>> file	make file as the standard output, append to it if it exists
<< word	take the shell input up to the first line containing ‘*word*’
command1 | command2	make the output of command1 as the input to command 2

IO Redirection

- Now try to make out what the following commands achieve



Commands	What does the command do?
$ ls > filelist	
$ date ; who > op	
$ date ; who; ls >op	
$ (date ; who) > op	
$ date; (who ; ls) > op	

Filters

- Many UNIX programs read some input, perform a transformation on it, and write it to some output

- – These programs are called *filters* and when used together with pipes can produce powerful programs

Filters

- Create a file named ***firstwill*** containing the following text:
- 1. Creativity is essentially a lonely art. ↲
- 2. An even lonelier struggle. ↲
- 3. To some a blessing. ↲
- 4. To others a curse. ↲
- 5. It is in reality the ability to reach inside yourself and drag forth
- from your very soul an idea. ↲
- 6. There is something that is much more scarce. ↲
- 7. Something rarer than ability. ↲
- 8. It is the ability to recognize ability. ↲
- 9. The purpose of life is a life of purpose. ↲
- 10. Life is simple, it's just not easy. ↲

Filters WC

- **wc**
- **Syntax: wc [OPTION] ... [file name]**
- [SysPgm@localhost ~] $ wc firstwill↵
- Output: 11 82 429 firstwill
- This means the file ***firstwill*** has 11 lines, 82 words and 429 characters
- options –l, -w, -c to get the number of lines, words characters individually
- [SysPgm@labserver ~] $ wc –l firstwill↵
- [SysPgm@labserver ~] $ wc –w firstwill↵
- [SysPgm@labserver ~] $ wc –c firstwill↵

Filters sort

- sort
- • Create a file “finalwill” with following contents
- 9
- 8
- 6
- 7
- 5
- 4
- 2
- 3
- 1
- sort command is used to sort the contents of the file

Filters sort

- **Syntax: sort [OPTION] ... [file name]...**
- While sorting the files
- – sort command compares the first character in each of the lines
- – If the first character for two lines is same then the second character is used in comparison
- [SysPgm@labserver ~] $ sort finalwill↲

Filters sort

- [SysPgm@labserver ~] $ sort finalwill↵
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- sort is used to sort the contents of more than one file at a time

Filters sort

- Create a file named “finalwish” with following contents
- 24
- 22
- 4
- 28
- 14
- 32
- 9
- 21
- 61
- 27
- Execute the command for sorting 2 files “finalwill” and “finalwish”

Filters sort

```
• [SysPgm@labserver ~] $ sort finalwill finalwish↵
• 1
• 14
• 2
• 21
• 22
• 24
• 27
• 28
• 3
• 32
• 4456
• 61
• 7
• 8
• 9
• 9
```

Filters sort

- Execute
- [SysPgm@labserver ~] $ sort firstwill finalwill↲1
- Check the Out Put

Filters sort

- If we want the repeated lines to be ignored, we can use –u option with the sort command.
- The most common flags are as follows:
- ***Option Comment***
- **-b** Ignore leading blanks
- **-d** Consider only blanks and alphanumeric characters
- **-f** Fold lowercase to uppercase characters before sorting (i.e.,"Bill","bill" and “BILL” are treated the same)
- **-r** Reverse the result of comparisons

Filters head

- **head**
- – Reads the first few lines of any text given to it as an input and writes them to the display screen
- – By default, head returns the first ten lines of each file name that is provided to it
- **Syntax:** head [options] [file(s)]
- [SysPgm@labserver ~] $ head firstwill↲
- – displays the first ten lines of the file named “firstwill”

Filters head

- [SysPgm@labserver ~] $ head firstwill finalwill↵
- ==> firstwill <==
- 1. Creativity is essentially a lonely art.
- 2. An even lonelier struggle.
- 3. To some a blessing.
- 4. To others a curse.
- 5. It is in reality the ability to reach inside yourself and drag forth from your very soul an
- idea.
- 6. There is something that is much more scarce.
- 7. Something rarer than ability.
- 8. It is the ability to recognize ability.
- 9. The purpose of life is a life of purpose.
- 10. Life is simple, it's just not easy.
- ==> finalwill <==

Contents of finalwill file

Filters head

- We can specify the number of lines to be displayed from the file by using –n option.
- – The *-n* option is used followed by an integer indicating the number of lines desired.
- For example, if 5 lines of “firstwill” and “finalwill” to be displayed then

- [SysPgm@labserver ~] $ head –n5 firstwill finalwill↵

Filters head

```
• [SysPgm@localhost ~] $ head –n5 firstwill finalwill↵
• ==> firstwill <==
• 1. Creativity is essentially a lonely art.
• 2. An even lonelier struggle.
• 3. To some a blessing.
• 4. To others a curse.
• 5. It is in reality the ability to reach inside yourself and drag forth
  from your very
• soul an idea.
• ==> finalwill <==
• 98675
```

Filters tail

- tail
- – similar to the head command except that it reads the final lines in files rather than the first lines
- [SysPgm@localhost ~] $ tail –n5 firstwill finalwill↵
- ==> firstwill <==
- 7. Something rarer than ability.
- 8. It is the ability to recognize ability.
- 9. The purpose of life is a life of purpose.
- 10. Life is simple, it's just not easy.
- ==> finalwill <==
- 4231

Filters grep

- grep
- – helps in searching strings in a file
- Syntax: **grep "string" FILE_PATTERN**
- Create a file named ***food*** with the following contents:
- *Afghani Cuisine, Mumbai, India*
- *Mandalay*
- *Big Apple Deli*
- *Isle of Java*
- *Tio Pepe's Peppers*
- *Sushi and Sashimi*
- *Sweet Tooth*
- *Bangkok Wok*

Filters grep

- To find the lines that contain the search string we use grep
- ***Example: to find the string "Wok"*** from the file "food" and display those lines to the standard output; we use the following command:
- [SysPgm@labserver ~] $ grep Wok food↵
- – displays the line(s) containing ***Wok*** in the file ***food***
- – It is done by matching the pattern ***Wok*** in the file ***food***

Filters grep

- [SysPgm@localhost ~] $ cat food | grep Wok↵
- Find the output
- grep acts as a filter here
- Checking for a given string in multiple files can be done with the same syntax as in
- **grep "string" FILE_PATTERN**
- grep output will include the file name in front of the line that matches the specific pattern
- – When the Linux shell sees the meta-character, it does the expansion and gives all the files as input to grep.

Filters grep

- grep options
- – To find how many lines matches the pattern, use –c option as follows
- grep -c FILE_PATTERN
- – To find string irrespective of case, use –i option as follows
- grep – i "string" FILE_PATTERN

Filters pipe

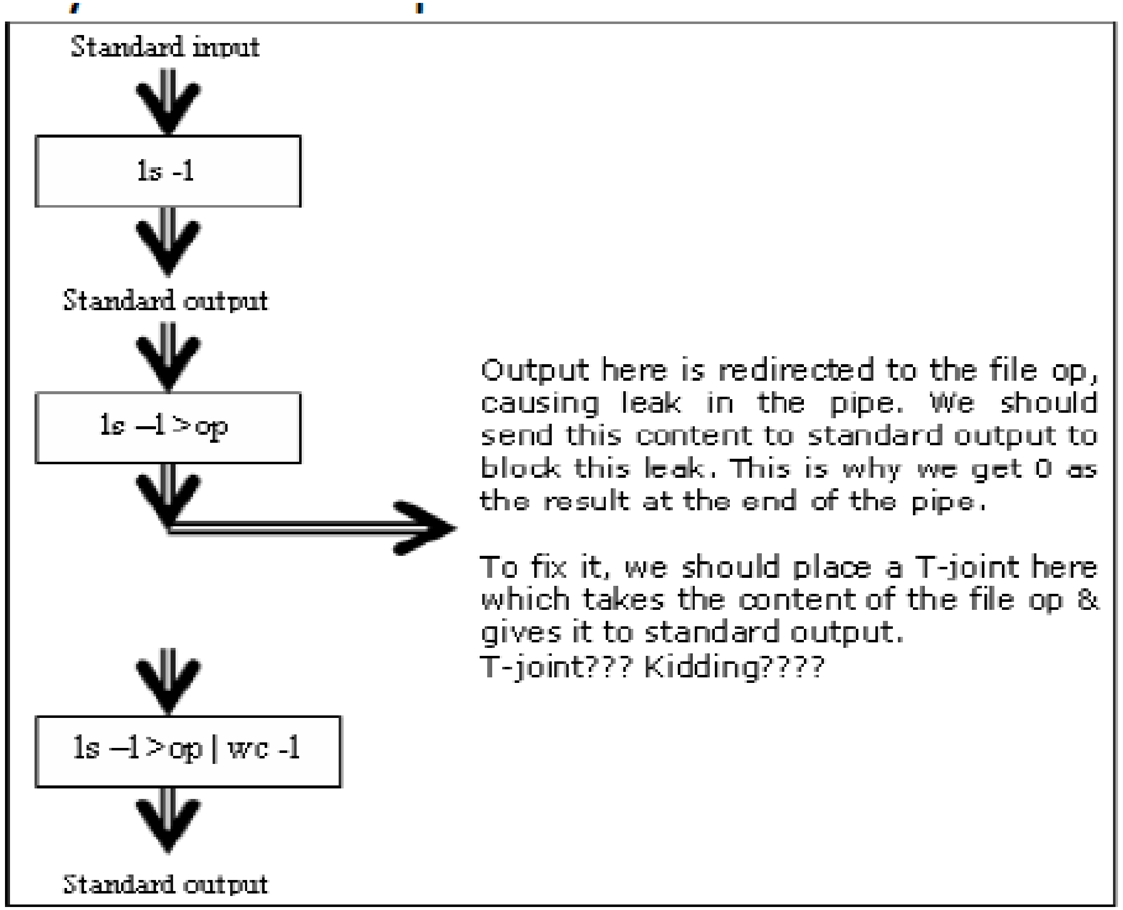
- piping
- – The indirection operator (**|**) is called as pipe symbol
- – helps joining two commands
- – output from one command becomes the input of the command
- – the standard output of the command to the left of the pipe symbol becomes the standard input of the command to the right of the pipe symbol
- – Two or more commands connected in this way forms a **pipe**

Filters pipe

- [SysPgm@labserver ~] $ ls | wc↵
- – Will execute the ls command first, direct the output to the next command wc
- – The command wc is executed and the output is shown in the terminal
- [SysPgm@labserver ~] $ cat finalwill| sort↵
- [SysPgm@labserver ~] $ ls > op| wc↵

Filters pipe

- Why is the output zero?

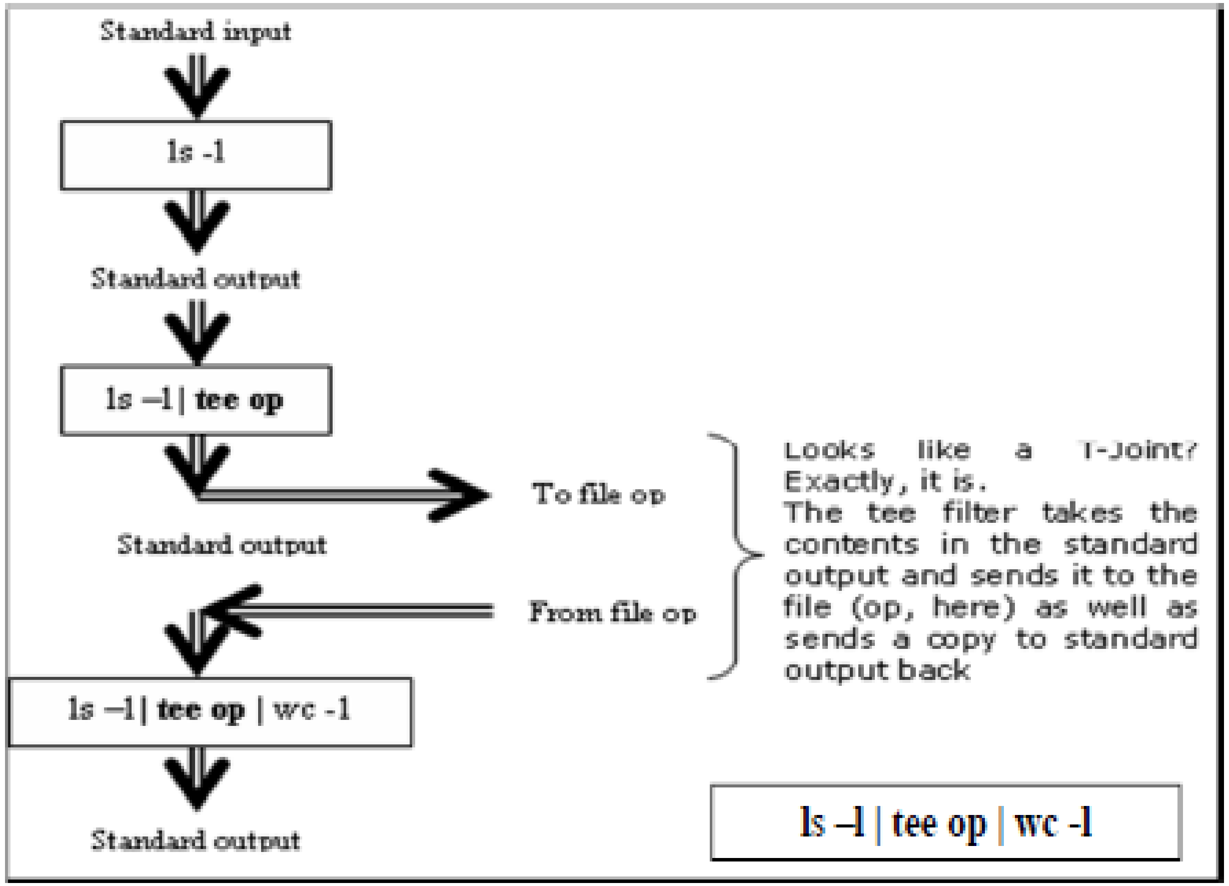


Output here is redirected to the file op, causing leak in the pipe. We should send this content to standard output to block this leak. This is why we get 0 as the result at the end of the pipe.

To fix it, we should place a T-joint here which takes the content of the file op & gives it to standard output.
T-joint??? Kidding????

Filters pipe

- Repaired the pipe with tee joint



Looks like a T-Joint? Exactly, it is. The tee filter takes the contents in the standard output and sends it to the file (op, here) as well as sends a copy to standard output back

```
ls –l | tee op | wc -l
```

Filters tee

- **tee**
- – is used to direct the output to a file as well as the
- standard output which is the terminal.
- Try [SysPgm@labserver ~] $ ls –l | tee op | wc –l ↲
- • It displays the number of lines (wc -l) in the
- directory listing (ls -l) on the terminal

THANK YOU