# Shell

By: Kshitij Gupta (2017B3A70601P)
Pranali Sancheti (2017B3A70736P)
Rahul Poddar(2017B3A70746P)

## ● How to run

1. To compile and execute the shell program, use the Makefile using the command → make.

## ● Features Implemented

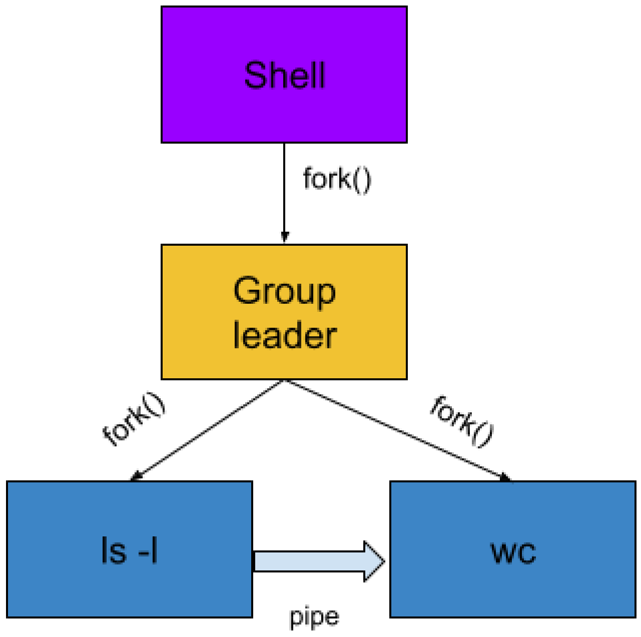
Following features are supported by the shell :

1) Pipelining of commands
   ls|wc|wc|wc
2) Input and output redirection
   ls > result.txt
   ls >> result.txt
   wc < a1.txt
   (Please give a space between the command and the redirection arrows for correct execution of the command).
3) Input and output redirection combined with pipes
   ls | wc > output.txt
4) Background and Foreground processes

| | |
|---|---|
| ls \| wc & | Background process |
| ls \| wc | Foreground process |

5) Implemented a short-cut command. In this mode, a command can be executed by pressing Ctrl-C and pressing a number. This number corresponds to the index in the lookup table created and deleted by the commands sc -i <index> <cmd>/ sc -d <index> <cmd>.
   sc -i 50 ls | wc (inserting command in lookup table)
   Ctrl-C + 50 (executing command)
   sc -d 50 ls | wc (deleting command from the lookup table)

6) Two new pipeline operators "||" and "|||" are implemented in shell.
   ls -l || grep ^-, grep ^d
   In double pipe, output of one command (ls -l) command is passed as input to two other commands.
   ls -l ||| grep ^-, grep ^d, wc
   In triple pipe,output of one command is passed as input to three other commands
7) For every command entered the Shell creatings a new process group.



- Working

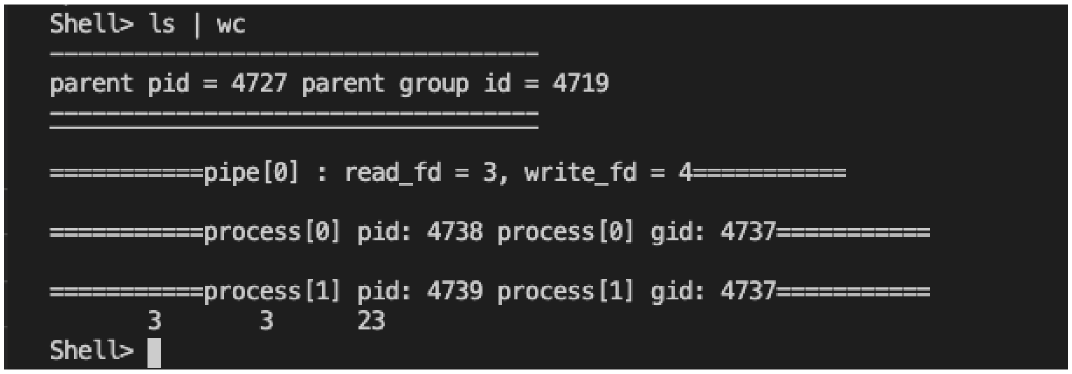
1) **Taking input :** For every input, the shell creates a child process and makes it a group leader. All the execution work is done by this child process and the shell is ready for the next input.
If the input command is of the form sc -i <index> <cmd>, then the parsing of command is done by the  parser function , and the linked list of command structure generated as result is stored in the lookup table.
If the input command is of the form,sc -d <index> <cmd>, then the following entry corresponding to index is removed from the lookup table.
A signal handler is created to handle SIGINT signal, generated by pressing Ctrl+C. The signal handle function asks the user for a number. The number is searched in the lookup table, which is implemented as a hashing with chaining. Then the corresponding entry in the lookup table is executed, if the number is found.

2) **Parsing :** The command read from the shell is tokenized with '|' (pipe) as our tokenizer. Each individual token is then further tokenized with ',' (comma) as the tokenizer, this is mainly done to accommodate the double-pipe (||) and triple-pipe(|||) feature. Now, for each of these tokens, a command structure is initialised and is added as a node to the

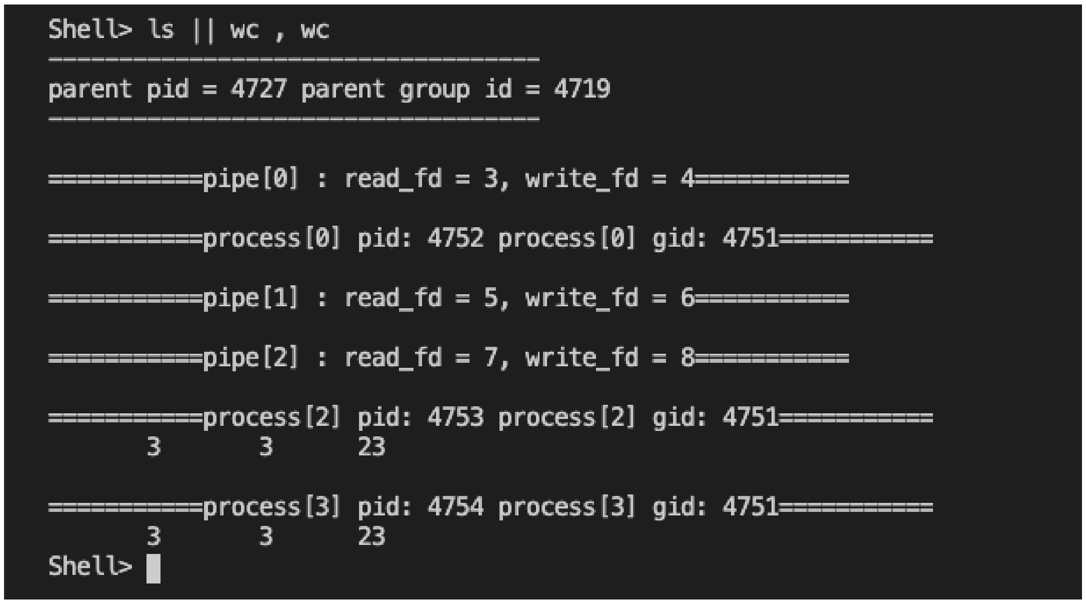
linked list of commands. If we encounter the symbols '>' , '<' , '>>' during parsing we set the corresponding flags and store the file name.

3) **Execution :** We start by iterating the linked list and creating a child process for every non-empty node. The command corresponding to every node is then executed by passing it to the execv() function call. We also create pipes for every consecutive pair of nodes. These pipes are used for communicating the output of the first node to the second, which uses it as input during execution. If the flags for either of '>' , '<' , '>>' are set for a node we change the file descriptors accordingly.

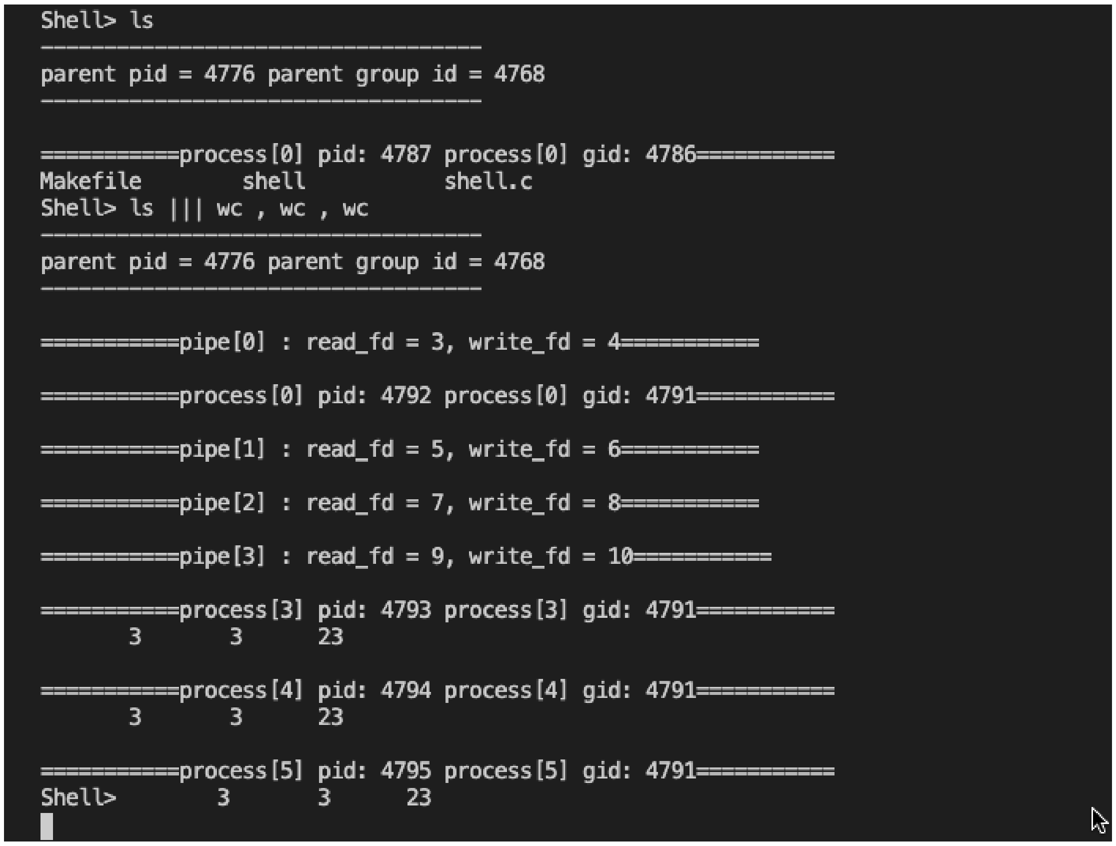
Screenshots



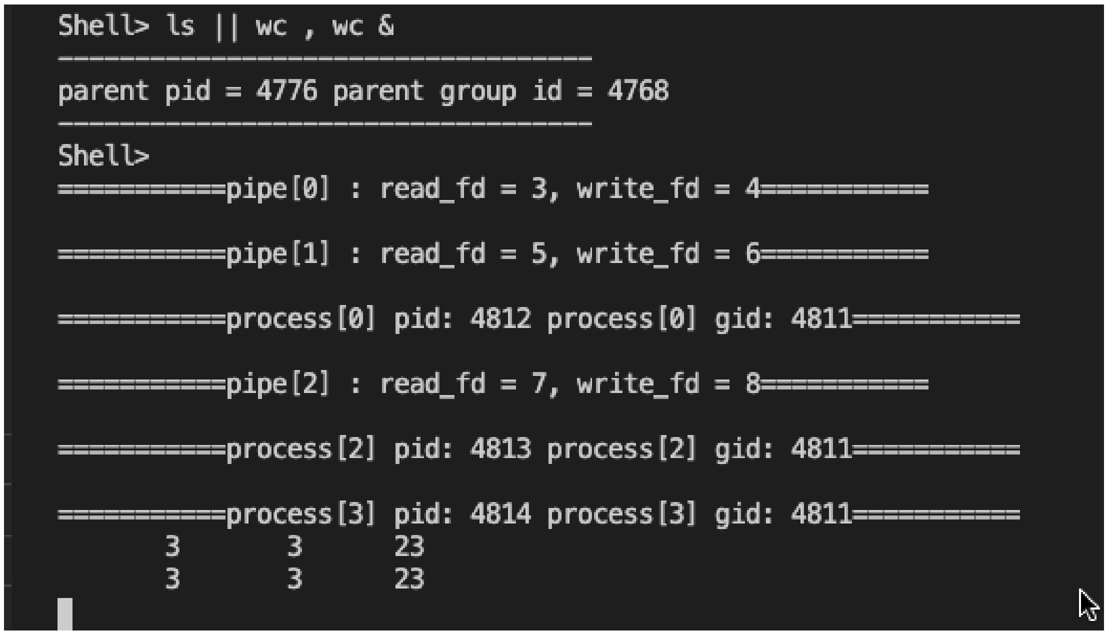
```
Shell> ls | wc
-----------------------------------
parent pid = 4727 parent group id = 4719
-----------------------------------

===========pipe[0] : read_fd = 3, write_fd = 4===========

===========process[0] pid: 4738 process[0] gid: 4737===========

===========process[1] pid: 4739 process[1] gid: 4737===========
       3       3      23
Shell> 
```



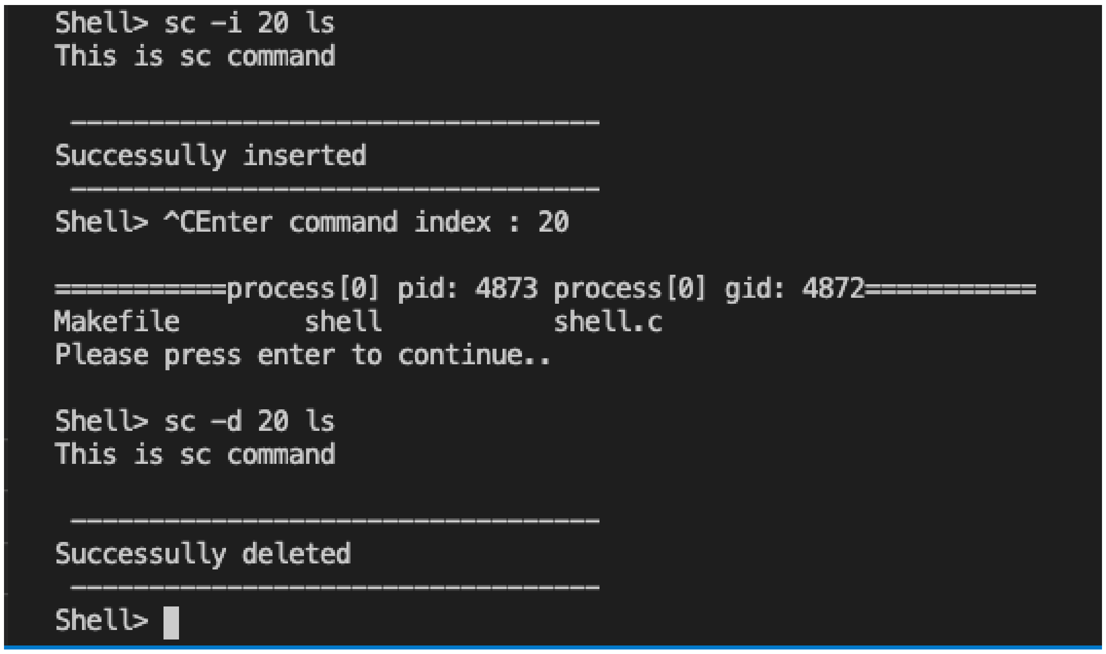
```
Shell> ls || wc , wc
-----------------------------------
parent pid = 4727 parent group id = 4719
-----------------------------------

===========pipe[0] : read_fd = 3, write_fd = 4===========

===========process[0] pid: 4752 process[0] gid: 4751===========

===========pipe[1] : read_fd = 5, write_fd = 6===========

===========pipe[2] : read_fd = 7, write_fd = 8===========

===========process[2] pid: 4753 process[2] gid: 4751===========
       3       3      23

===========process[3] pid: 4754 process[3] gid: 4751===========
       3       3      23
Shell> 
```



```
Shell> ls
-----------------------------------
parent pid = 4776 parent group id = 4768
-----------------------------------

===========process[0] pid: 4787 process[0] gid: 4786===========
Makefile        shell           shell.c
Shell> ls ||| wc , wc , wc
-----------------------------------
parent pid = 4776 parent group id = 4768
-----------------------------------

===========pipe[0] : read_fd = 3, write_fd = 4===========

===========process[0] pid: 4792 process[0] gid: 4791===========

===========pipe[1] : read_fd = 5, write_fd = 6===========

===========pipe[2] : read_fd = 7, write_fd = 8===========

===========pipe[3] : read_fd = 9, write_fd = 10===========

===========process[3] pid: 4793 process[3] gid: 4791===========
       3       3      23

===========process[4] pid: 4794 process[4] gid: 4791===========
       3       3      23

===========process[5] pid: 4795 process[5] gid: 4791===========
Shell>        3       3      23
```



```
Shell> ls || wc , wc &
-----------------------------------
parent pid = 4776 parent group id = 4768
-----------------------------------
Shell>
===========pipe[0] : read_fd = 3, write_fd = 4===========

===========pipe[1] : read_fd = 5, write_fd = 6===========

===========process[0] pid: 4812 process[0] gid: 4811===========

===========pipe[2] : read_fd = 7, write_fd = 8===========

===========process[2] pid: 4813 process[2] gid: 4811===========

===========process[3] pid: 4814 process[3] gid: 4811===========
       3       3      23
       3       3      23
```



```
Shell> sc -i 20 ls
This is sc command

 ----------------------------------
Successully inserted
 ----------------------------------
Shell> ^CEnter command index : 20

===========process[0] pid: 4873 process[0] gid: 4872===========
Makefile        shell           shell.c
Please press enter to continue..

Shell> sc -d 20 ls
This is sc command

 ----------------------------------
Successully deleted
 ----------------------------------
Shell> 
```