

SKSHELL Shell Program Manual

Author : SREENIDHI KRISHNA

Date : 10/06/2014

SUMMARY

This simple shell program written in C language for UNIX/LINUX systems can handle basic terminal commands and includes the following features:

1. Basic command : <command> <flags>
2. Output Redirection : <command> <flags> > <filename>
3. Input Redirection : <command> <flags> < <filename>
4. Piping : <command> <flags> | <command> <flags>
5. Background : <command> <flags> &
6. Exit the shell : exit

Note : (a) Full path needs to be specified for all the commands

(b) A single whitespace acts as delimiter between all commands, arguments and special symbols

PROGRAM MODULES

1. int main(int argc, char *argv[])

The Main function of the shell takes input from the user and passes it to parsecmd function for command and argument parsing. The main function then checks for the occurrence of special arguments such as "<" for input redirection, ">" for output redirection, "|" for piping and "&" for initiating background processes. It also creates a pipe and uses fork function for the creation of child process. Shell termination is achieved using the command "exit".

2. void parsecmd(char *,char **,int *)

The input string is passed to the parsecmd function for command parsing. The parsecmd makes use of the function strtok() to create string tokens for each command and argument.

3.void runpipe(int [], char **, char **)

The runpipe function is evoked to carry out the piping between two processes. It forks and creates a child and parent process to implement the pipe. The redirection of pipe output and stdout is carried out in parent and the redirection of pipe input and stdin is carried out in child process.

INSTALLATION & TESTING

To build SKSHELL, use the makefile included in the distribution and type

\$ make

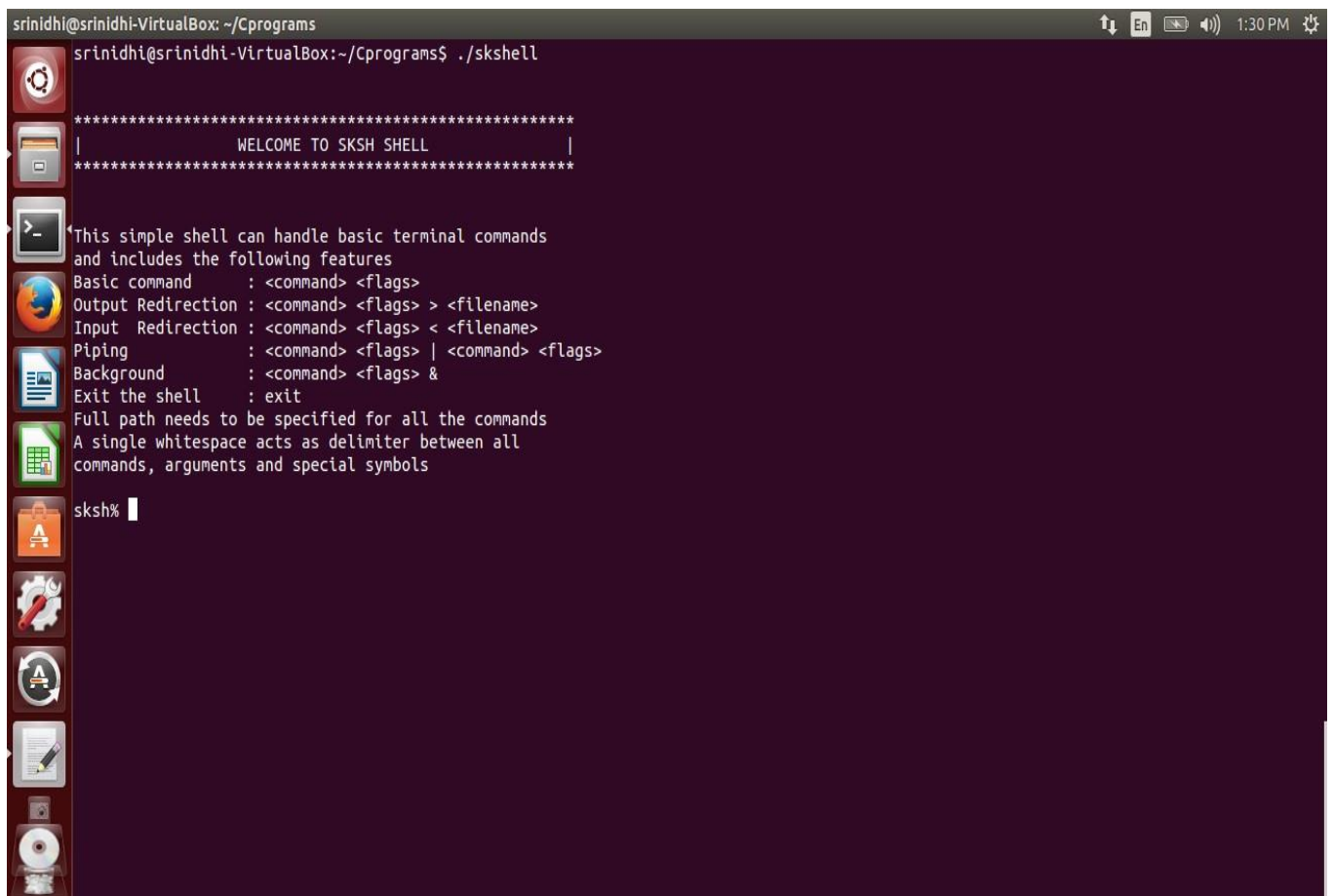
To run the program, type

\$./skshell

SAMPLE OUTPUTS

The features of the SKSH shell program is demonstrated in the following screen caps.

1. Shell Initialization



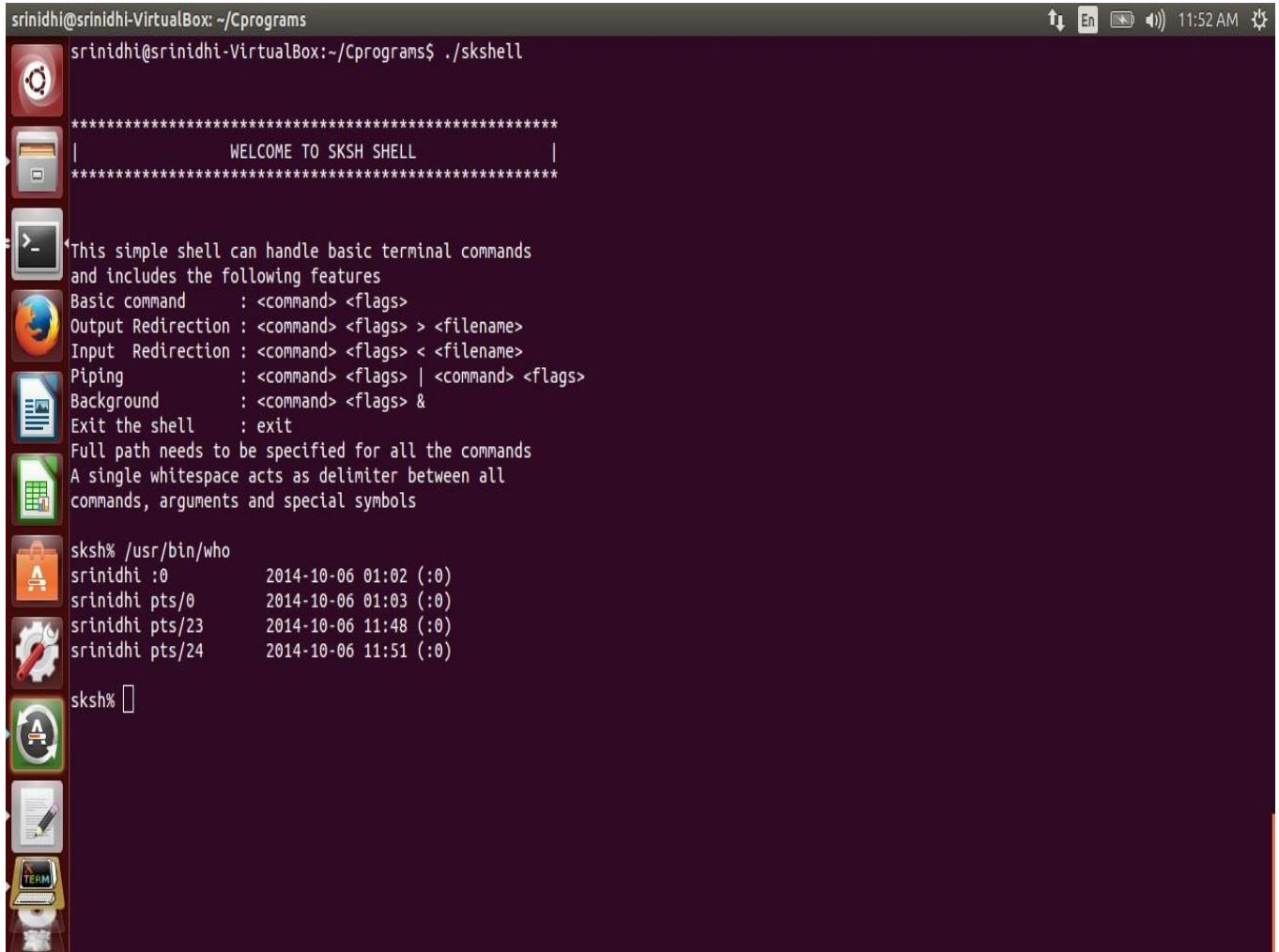
```
srinidhi@srinidhi-VirtualBox: ~/Cprograms
srinidhi@srinidhi-VirtualBox:~/Cprograms$ ./skshell
*****
|                               |
|         WELCOME TO SKSH SHELL         |
|                               |
*****

This simple shell can handle basic terminal commands
and includes the following features
Basic command      : <command> <flags>
Output Redirection : <command> <flags> > <filename>
Input Redirection  : <command> <flags> < <filename>
Piping             : <command> <flags> | <command> <flags>
Background         : <command> <flags> &
Exit the shell     : exit
Full path needs to be specified for all the commands
A single whitespace acts as delimiter between all
commands, arguments and special symbols

sksh% 
```

2. Basic Command Execution eg. /usr/bin/who

This screenshot illustrates the execution of a simple command



```
srinidhi@srinidhi-VirtualBox: ~/Cprograms
srinidhi@srinidhi-VirtualBox:~/Cprograms$ ./skshell
*****
|                               |
|      WELCOME TO SKSH SHELL    |
|                               |
*****

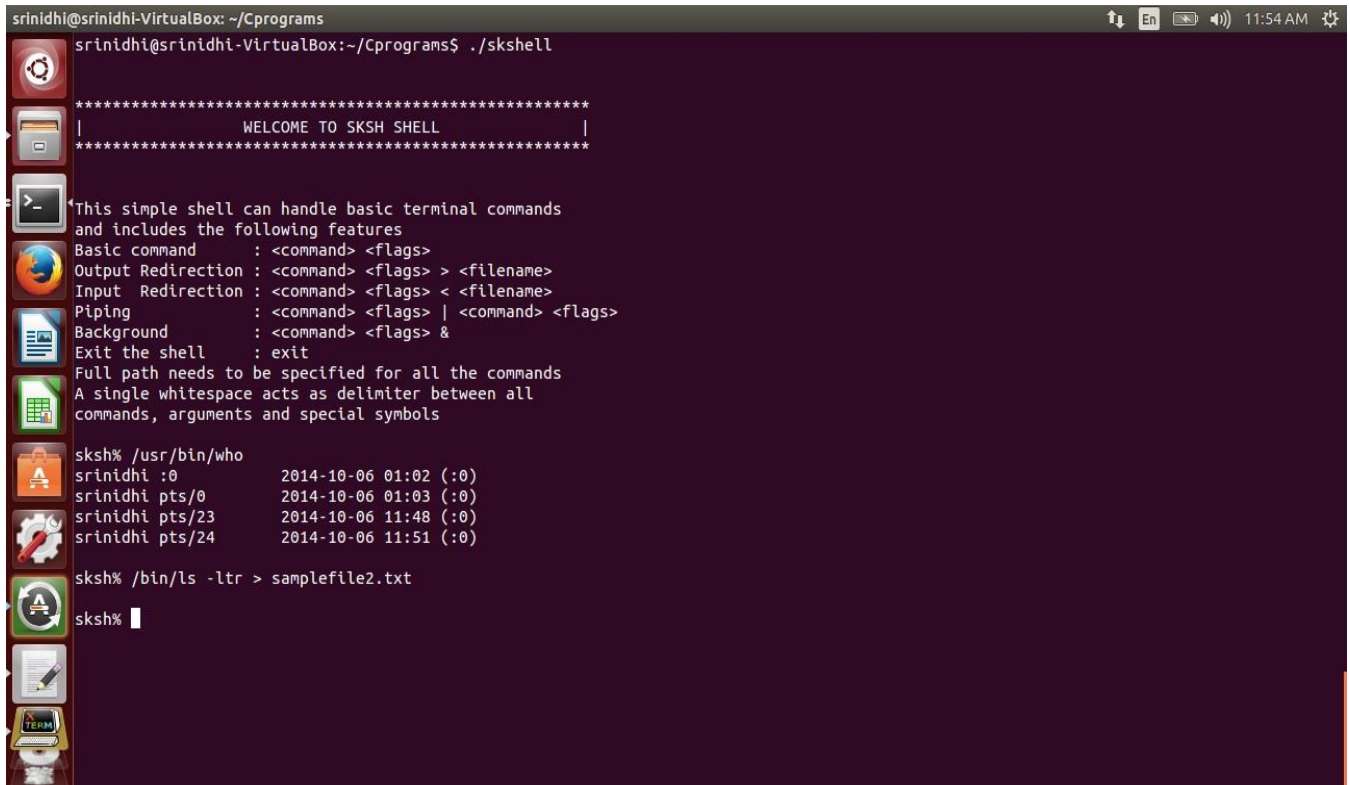
This simple shell can handle basic terminal commands
and includes the following features
Basic command      : <command> <flags>
Output Redirection : <command> <flags> > <filename>
Input Redirection  : <command> <flags> < <filename>
Piping             : <command> <flags> | <command> <flags>
Background         : <command> <flags> &
Exit the shell     : exit
Full path needs to be specified for all the commands
A single whitespace acts as delimiter between all
commands, arguments and special symbols

sksh% /usr/bin/who
srinidhi :0          2014-10-06 01:02 (:0)
srinidhi pts/0      2014-10-06 01:03 (:0)
srinidhi pts/23     2014-10-06 11:48 (:0)
srinidhi pts/24     2014-10-06 11:51 (:0)

sksh% 
```

3. Output Redirection eg. `/bin/ls -ltr > samplefile2.txt`

This screenshot illustrates output redirection. The output of the command is written into the file.



```
srinidhi@srinidhi-VirtualBox: ~/Cprograms
srinidhi@srinidhi-VirtualBox:~/Cprograms$ ./skshell

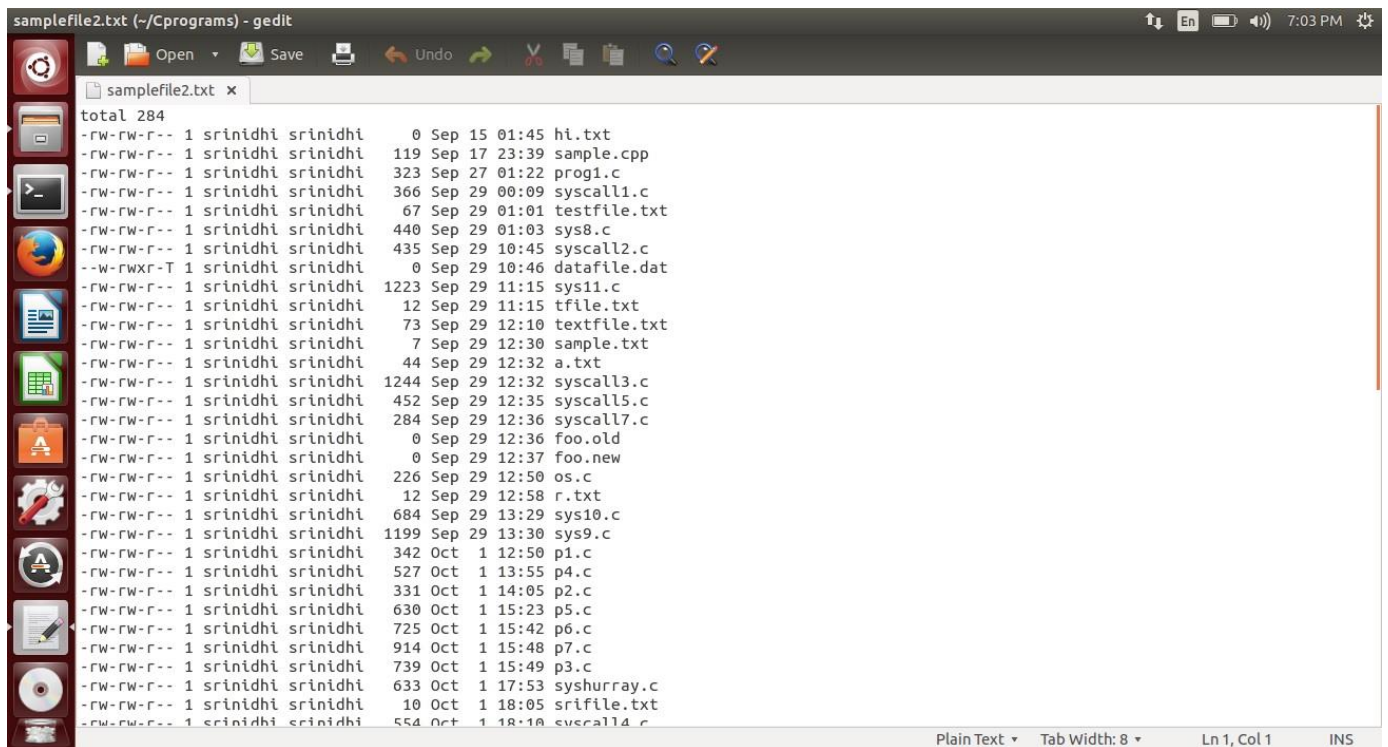
*****
|               WELCOME TO SKSH SHELL               |
*****

This simple shell can handle basic terminal commands
and includes the following features
Basic command      : <command> <flags>
Output Redirection : <command> <flags> > <filename>
Input Redirection  : <command> <flags> < <filename>
Piping             : <command> <flags> | <command> <flags>
Background         : <command> <flags> &
Exit the shell     : exit
Full path needs to be specified for all the commands
A single whitespace acts as delimiter between all
commands, arguments and special symbols

sksh% /usr/bin/who
srinidhi :0          2014-10-06 01:02 (:0)
srinidhi pts/0      2014-10-06 01:03 (:0)
srinidhi pts/23     2014-10-06 11:48 (:0)
srinidhi pts/24     2014-10-06 11:51 (:0)

sksh% /bin/ls -ltr > samplefile2.txt

sksh%
```



```
samplefile2.txt (~/Cprograms) - gedit
total 284
-rw-rw-r-- 1 srinidhi srinidhi    0 Sep 15 01:45 hi.txt
-rw-rw-r-- 1 srinidhi srinidhi   119 Sep 17 23:39 sample.cpp
-rw-rw-r-- 1 srinidhi srinidhi   323 Sep 27 01:22 prog1.c
-rw-rw-r-- 1 srinidhi srinidhi   366 Sep 29 00:09 syscall1.c
-rw-rw-r-- 1 srinidhi srinidhi    67 Sep 29 01:01 testfile.txt
-rw-rw-r-- 1 srinidhi srinidhi   440 Sep 29 01:03 sys8.c
-rw-rw-r-- 1 srinidhi srinidhi   435 Sep 29 10:45 syscall2.c
--w-rwxr-T 1 srinidhi srinidhi     0 Sep 29 10:46 datafile.dat
-rw-rw-r-- 1 srinidhi srinidhi  1223 Sep 29 11:15 sys11.c
-rw-rw-r-- 1 srinidhi srinidhi    12 Sep 29 11:15 tfile.txt
-rw-rw-r-- 1 srinidhi srinidhi    73 Sep 29 12:10 textfile.txt
-rw-rw-r-- 1 srinidhi srinidhi     7 Sep 29 12:30 sample.txt
-rw-rw-r-- 1 srinidhi srinidhi    44 Sep 29 12:32 a.txt
-rw-rw-r-- 1 srinidhi srinidhi  1244 Sep 29 12:32 syscall3.c
-rw-rw-r-- 1 srinidhi srinidhi   452 Sep 29 12:35 syscall5.c
-rw-rw-r-- 1 srinidhi srinidhi   284 Sep 29 12:36 syscall7.c
-rw-rw-r-- 1 srinidhi srinidhi     0 Sep 29 12:36 foo.old
-rw-rw-r-- 1 srinidhi srinidhi     0 Sep 29 12:37 foo.new
-rw-rw-r-- 1 srinidhi srinidhi   226 Sep 29 12:50 os.c
-rw-rw-r-- 1 srinidhi srinidhi    12 Sep 29 12:58 r.txt
-rw-rw-r-- 1 srinidhi srinidhi   684 Sep 29 13:29 sys10.c
-rw-rw-r-- 1 srinidhi srinidhi  1199 Sep 29 13:30 sys9.c
-rw-rw-r-- 1 srinidhi srinidhi   342 Oct  1 12:50 p1.c
-rw-rw-r-- 1 srinidhi srinidhi   527 Oct  1 13:55 p4.c
-rw-rw-r-- 1 srinidhi srinidhi   331 Oct  1 14:05 p2.c
-rw-rw-r-- 1 srinidhi srinidhi   630 Oct  1 15:23 p5.c
-rw-rw-r-- 1 srinidhi srinidhi   725 Oct  1 15:42 p6.c
-rw-rw-r-- 1 srinidhi srinidhi   914 Oct  1 15:48 p7.c
-rw-rw-r-- 1 srinidhi srinidhi   739 Oct  1 15:49 p3.c
-rw-rw-r-- 1 srinidhi srinidhi   633 Oct  1 17:53 syshurray.c
-rw-rw-r-- 1 srinidhi srinidhi    10 Oct  1 18:05 srifile.txt
-rw-rw-r-- 1 srinidhi srinidhi   554 Oct  1 18:10 syscall4.c
```

4. Input Re direction eg. /usr/bin/tr a-z A-Z < samplefile2.txt

This screenshot illustrates input redirection. The input to the command is obtained from the file.

```

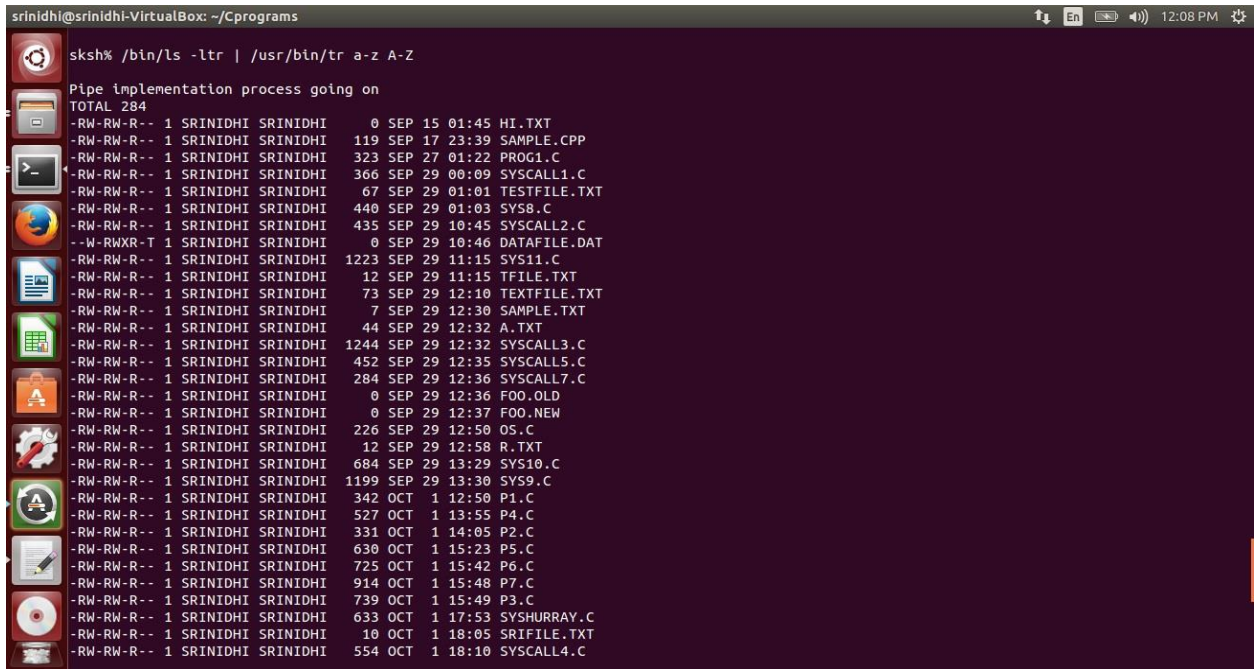
srinidhi@srinidhi-VirtualBox: ~/Cprograms
-rw-rw-r-- 1 SRINIDHI SRINIDHI 453 OCT 3 11:43 SS1.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1456 OCT 3 13:27 SS2.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 4149 OCT 3 17:05 FINAL.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1533 OCT 3 19:32 P8.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 442 OCT 4 12:36 B1.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 0 OCT 4 12:59 UNTITLED DOCUMENT 4
-rw-rw-r-- 1 SRINIDHI SRINIDHI 566 OCT 4 12:59 B2.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1335 OCT 4 13:23 B3.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1890 OCT 4 13:29 B4.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1526 OCT 4 14:53 B5.C
-rw-r--r-- 1 SRINIDHI SRINIDHI 38 OCT 4 15:08 DR.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 847 OCT 4 15:11 B6.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 986 OCT 4 15:38 B8.C
-rw-r--r-- 1 SRINIDHI SRINIDHI 1493 OCT 4 16:16 X.TXT
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1049 OCT 4 16:19 B7.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1048 OCT 4 16:22 B9.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 0 OCT 4 16:23 M.TXT
-rw-rw-r-- 1 SRINIDHI SRINIDHI 0 OCT 4 16:23 Y.TXT
-rw-rw-r-- 1 SRINIDHI SRINIDHI 0 OCT 4 16:25 Z.TXT
-rw-rw-r-- 1 SRINIDHI SRINIDHI 2337 OCT 4 18:14 MYPROGRAM.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1320 OCT 4 18:51 MY2.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1366 OCT 4 19:02 R.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 703 OCT 4 21:53 KILLPROG.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 2232 OCT 5 12:31 UNTITLED DOCUMENT 1
-rw-rw-r-- 1 SRINIDHI SRINIDHI 1294 OCT 5 15:23 K3.C
-rw-r--r-- 1 SRINIDHI SRINIDHI 1493 OCT 5 15:24 SAMPLE.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 0 OCT 5 15:30 SAMPLEFILE.TXT
-rw-rw-r-- 1 SRINIDHI SRINIDHI 88 OCT 5 17:31 SAMPLEFILE.TXT
-rw-rw-r-- 1 SRINIDHI SRINIDHI 4674 OCT 5 17:41 MYPROJ.C
-rw-rw-r-- 1 SRINIDHI SRINIDHI 542 OCT 5 18:20 TESTINGSTRING.C
-rwxrwxr-x 1 SRINIDHI SRINIDHI 8764 OCT 5 18:20 A.OUT
-rw-rw-r-- 1 SRINIDHI SRINIDHI 137 OCT 6 11:25 MAKEFILE
-rw-rw-r-- 1 SRINIDHI SRINIDHI 8207 OCT 6 11:27 SKSHELL.C
-rwxrwxr-x 1 SRINIDHI SRINIDHI 16911 OCT 6 11:34 SKSHELL
-rw-rw-r-- 1 SRINIDHI SRINIDHI 4094 OCT 6 11:43 SAMPLEFILE2.TXT

sksh%

```


5. Piping eg. `/bin/ls -ltr | /usr/bin/tr a-z A-Z`

This screenshot illustrates piping between two processes. The output of the first process serves as the input of the second process.



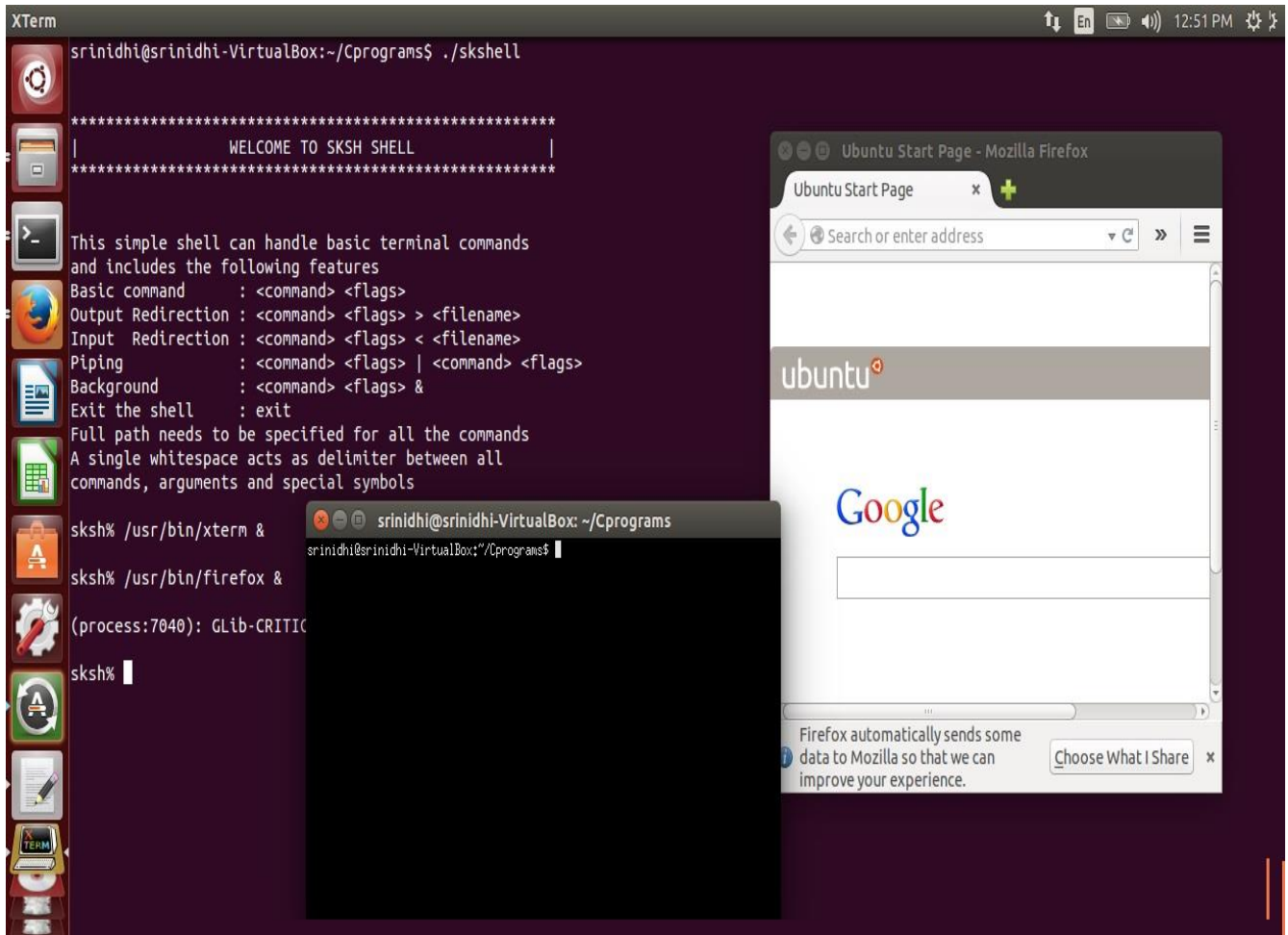
```
srinidhi@srinidhi-VirtualBox: ~/Cprograms
sksh% /bin/ls -ltr | /usr/bin/tr a-z A-Z

Pipe implementation process going on
TOTAL 284
-RW-RW-R-- 1 SRINIDHI SRINIDHI    0 SEP 15 01:45 HI.TXT
-RW-RW-R-- 1 SRINIDHI SRINIDHI   119 SEP 17 23:30 SAMPLE.CPP
-RW-RW-R-- 1 SRINIDHI SRINIDHI   323 SEP 27 01:22 PROG1.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   366 SEP 29 00:09 SYSCALL1.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI    67 SEP 29 01:01 TESTFILE.TXT
-RW-RW-R-- 1 SRINIDHI SRINIDHI   440 SEP 29 01:03 SYS8.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   435 SEP 29 10:45 SYSCALL2.C
--W-RWXR-T 1 SRINIDHI SRINIDHI    0 SEP 29 10:46 DATAFILE.DAT
-RW-RW-R-- 1 SRINIDHI SRINIDHI  1223 SEP 29 11:15 SYS11.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI    12 SEP 29 11:15 TFILE.TXT
-RW-RW-R-- 1 SRINIDHI SRINIDHI    73 SEP 29 12:10 TEXTFILE.TXT
-RW-RW-R-- 1 SRINIDHI SRINIDHI     7 SEP 29 12:30 SAMPLE.TXT
-RW-RW-R-- 1 SRINIDHI SRINIDHI    44 SEP 29 12:32 A.TXT
-RW-RW-R-- 1 SRINIDHI SRINIDHI  1244 SEP 29 12:32 SYSCALL3.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   452 SEP 29 12:35 SYSCALL5.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   284 SEP 29 12:36 SYSCALL7.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI     0 SEP 29 12:36 FOO.OLD
-RW-RW-R-- 1 SRINIDHI SRINIDHI     0 SEP 29 12:37 FOO.NEW
-RW-RW-R-- 1 SRINIDHI SRINIDHI   226 SEP 29 12:50 OS.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI    12 SEP 29 12:58 R.TXT
-RW-RW-R-- 1 SRINIDHI SRINIDHI   684 SEP 29 13:29 SYS10.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI  1199 SEP 29 13:30 SYS9.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   342 OCT  1 12:50 P1.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   527 OCT  1 13:55 P4.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   331 OCT  1 14:05 P2.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   630 OCT  1 15:23 P5.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   725 OCT  1 15:42 P6.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   914 OCT  1 15:48 P7.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   739 OCT  1 15:49 P3.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   633 OCT  1 17:53 SYSHURRAY.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI    10 OCT  1 18:05 SRIFILE.TXT
-RW-RW-R-- 1 SRINIDHI SRINIDHI   554 OCT  1 18:10 SYSCALL4.C
```

6. Background process eg. `/usr/bin/xterm &`

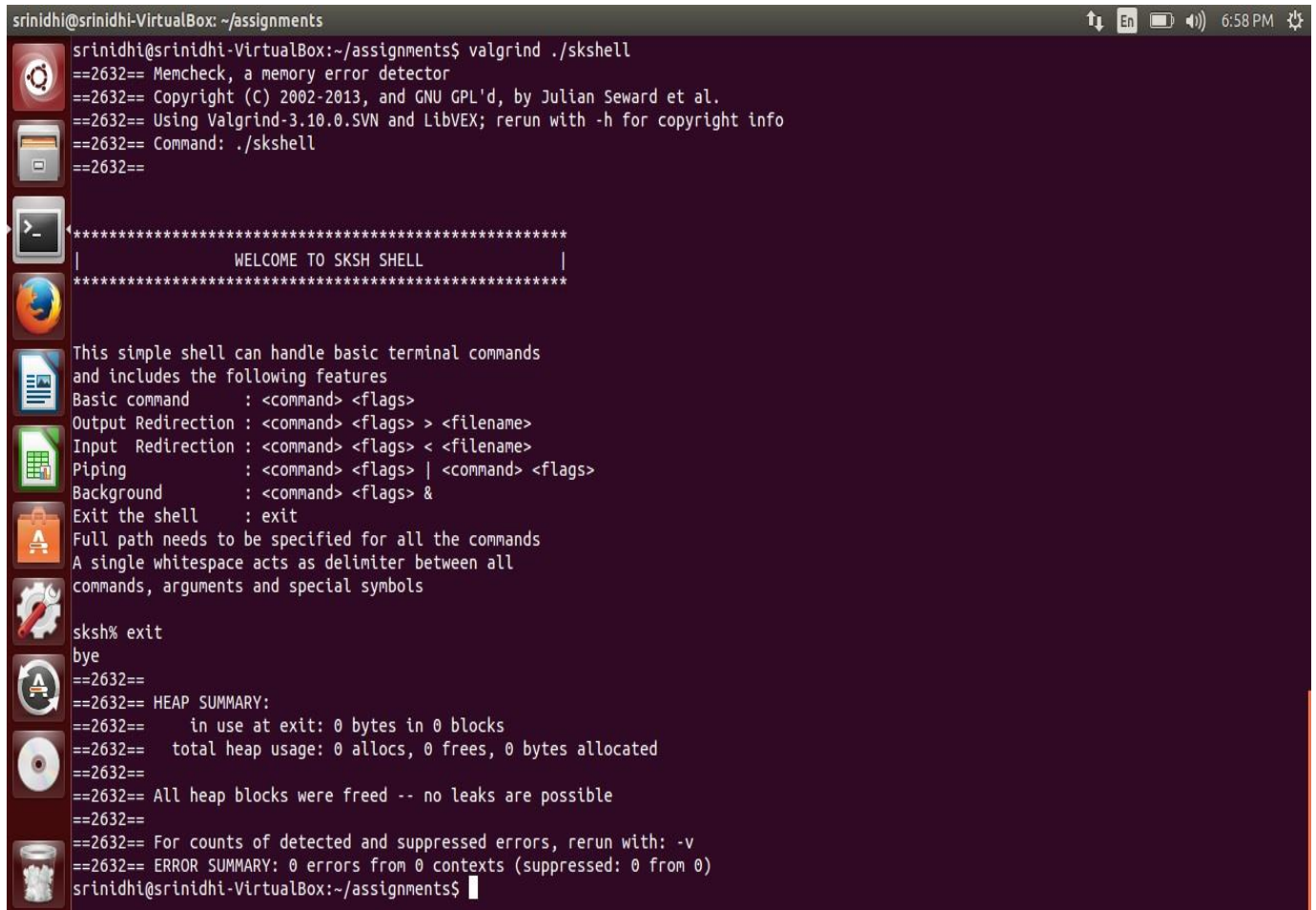
`/usr/bin/firefox &`

The following screenshot illustrates the execution of background processes.



7. Shell termination E.g: exit

The following screenshot demonstrates shell termination. Valgrind was used to test for memory leaks. No memory leaks were detected in the program and shell terminates cleanly.



```
srinidhi@srinidhi-VirtualBox: ~/assignments
srinidhi@srinidhi-VirtualBox:~/assignments$ valgrind ./skshell
==2632== Memcheck, a memory error detector
==2632== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==2632== Using Valgrind-3.10.0.SVN and LibVEX; rerun with -h for copyright info
==2632== Command: ./skshell
==2632==

*****
|                               |
|          WELCOME TO SKSH SHELL          |
|                               |
*****

This simple shell can handle basic terminal commands
and includes the following features
Basic command      : <command> <flags>
Output Redirection : <command> <flags> > <filename>
Input Redirection  : <command> <flags> < <filename>
Piping             : <command> <flags> | <command> <flags>
Background         : <command> <flags> &
Exit the shell     : exit
Full path needs to be specified for all the commands
A single whitespace acts as delimiter between all
commands, arguments and special symbols

sksh% exit
bye
==2632==
==2632==  HEAP SUMMARY:
==2632==    in use at exit: 0 bytes in 0 blocks
==2632==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==2632==
==2632== All heap blocks were freed -- no leaks are possible
==2632==
==2632== For counts of detected and suppressed errors, rerun with: -v
==2632== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
srinidhi@srinidhi-VirtualBox:~/assignments$
```