Design Description

Program: skshell.c

This simple shell can handle basic terminal commands and includes the following features

1. Basic command : <command> <flags>

2. Output Re direction : <command> <flags> > <filename>

3. Input Redirection : <command> <flags> < <filename>

4. Piping : <command> <flags> | <command> <flags>

5. Background : <command> <flags>

6. Exit the shell : exit

7. Full path needs to be specified for all the commands

8. A single whitespace acts as delimiter between all commands, arguments and special symbols

//Features:

Make

The various functions used in the program are

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 parsing.

The main function then checks for the occurence of various arguments such as "<" for input re direction , ">" for output re direction , "|" for piping function and "&" for initiating background processes. It also creates a pipe and uses fork function for the creation of child process.

The exit from the shell can done using the command "exit".

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

The input string is passed to the parsecmd function for parsing. The parsecmd makes use of the function strtok() for parsing.

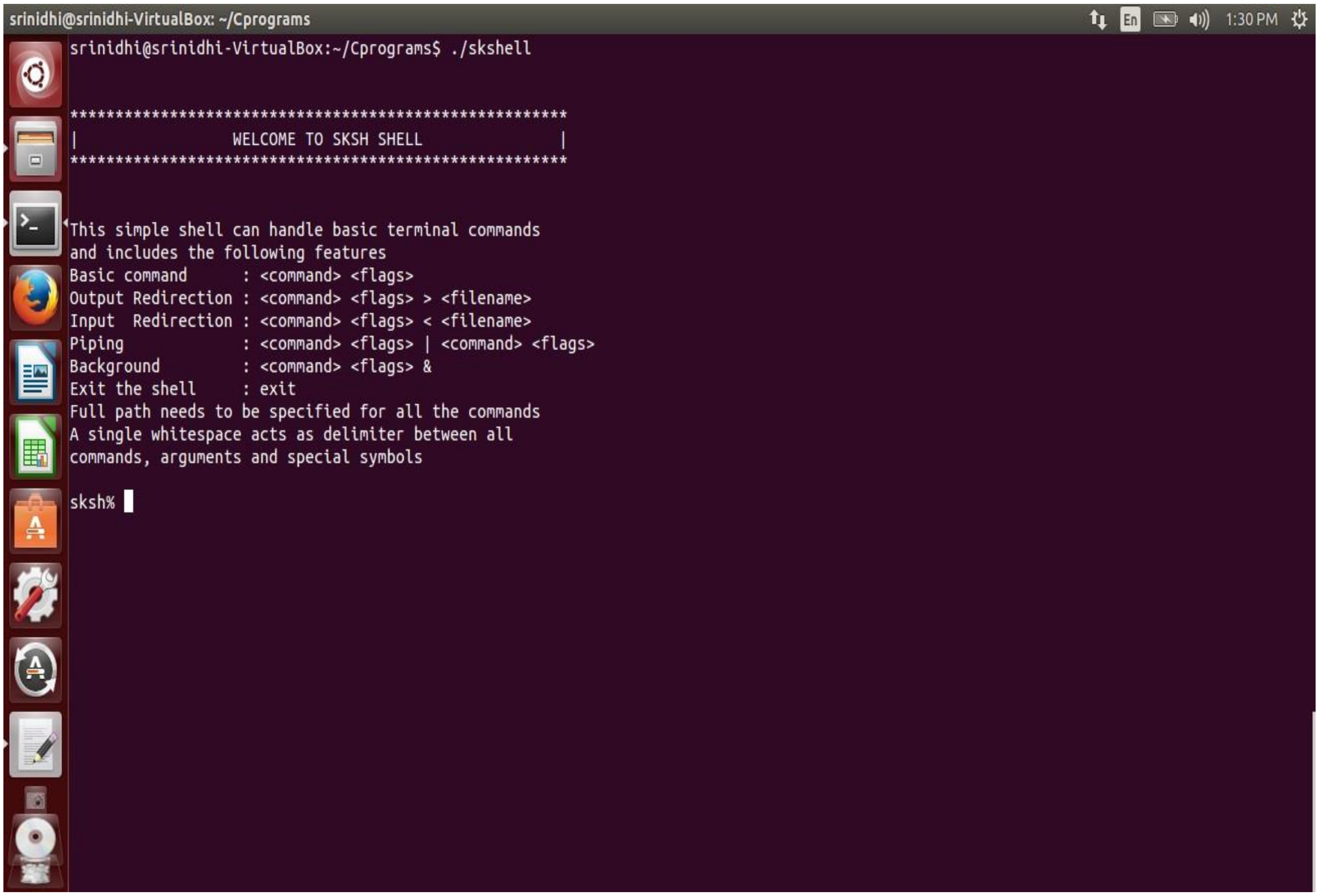
3.void runpipe(int [], char **, char **)

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

Sample Output

The working of the SKSH shell and its features are shown below.

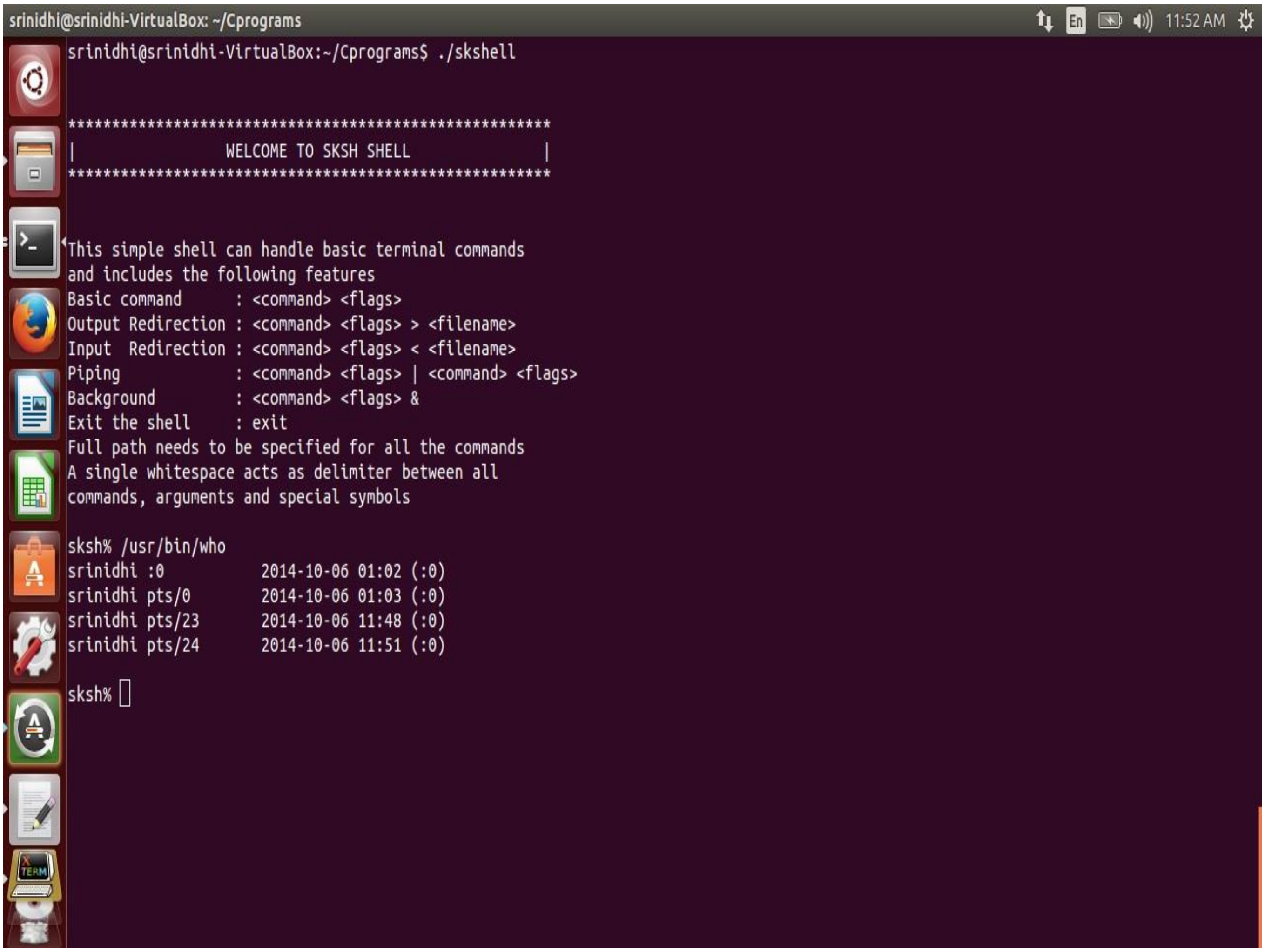
1. The Shell Structure:



```
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 eg. /usr/bin/who

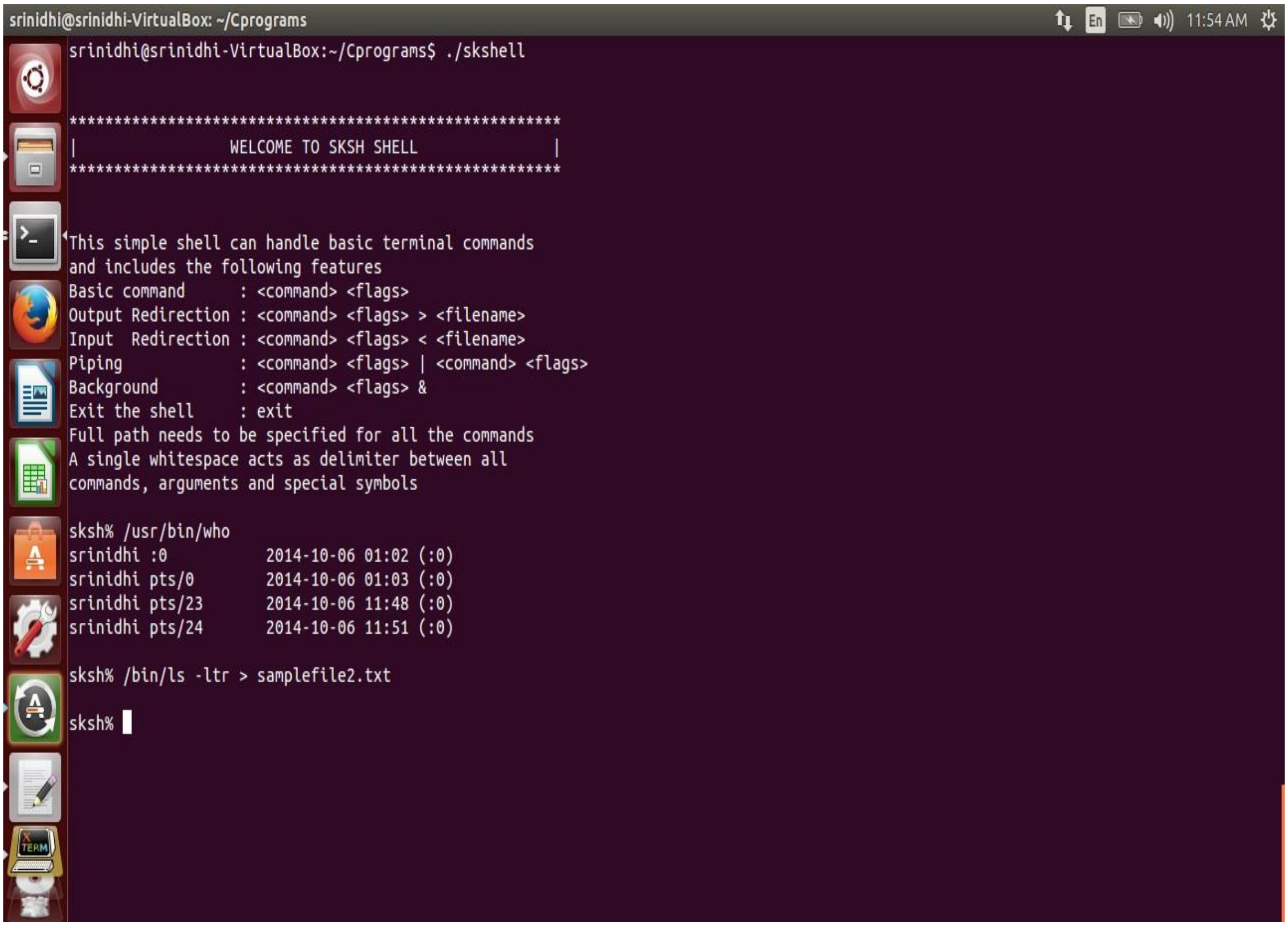
This illustrates the execution of a simple command



```
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 Re direction eg. /bin/ls - ltr > samplefile2.txt

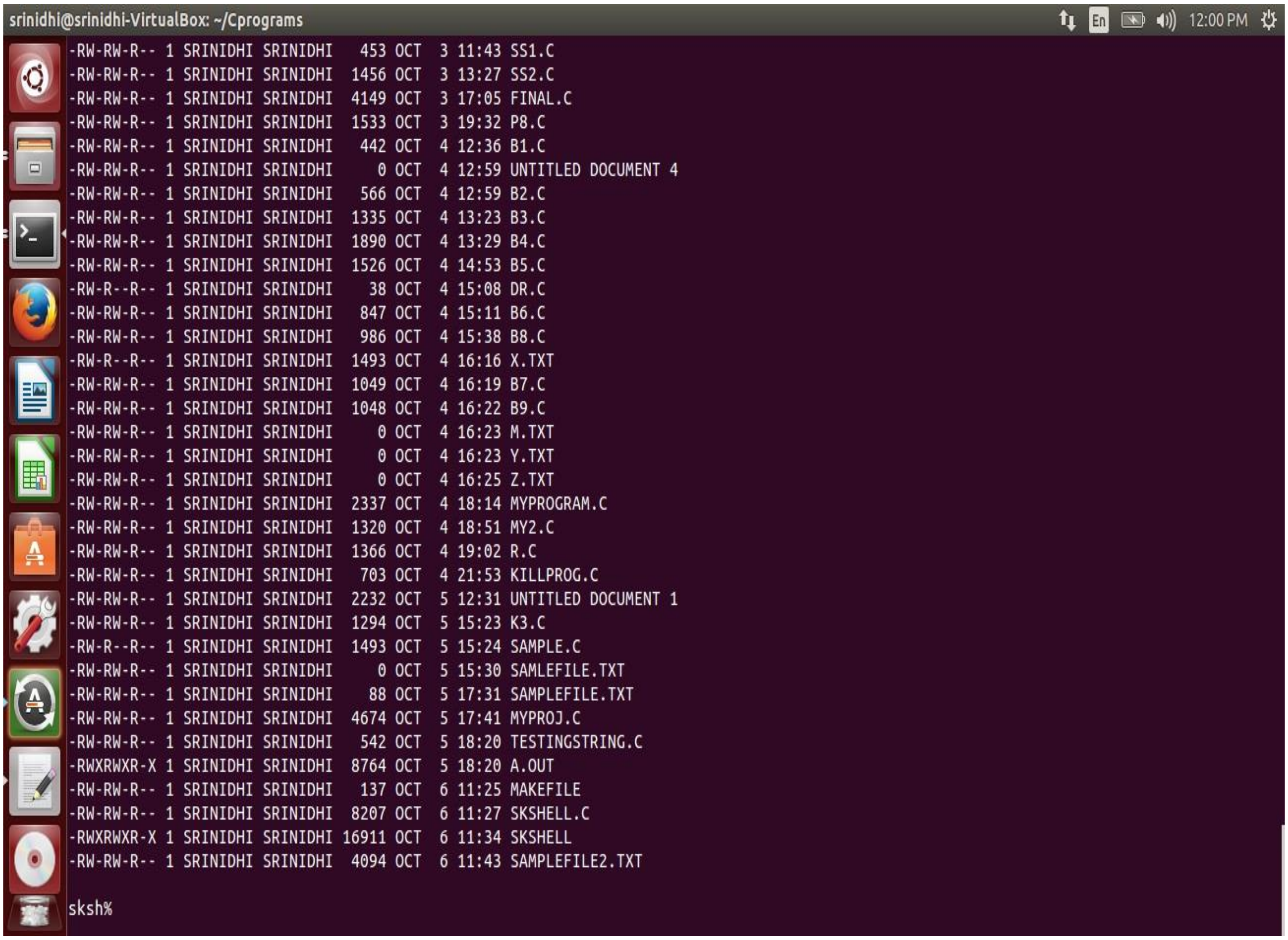
This illustrates the output re direction execution. The output of the command is written into the file.



```
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% 
```

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

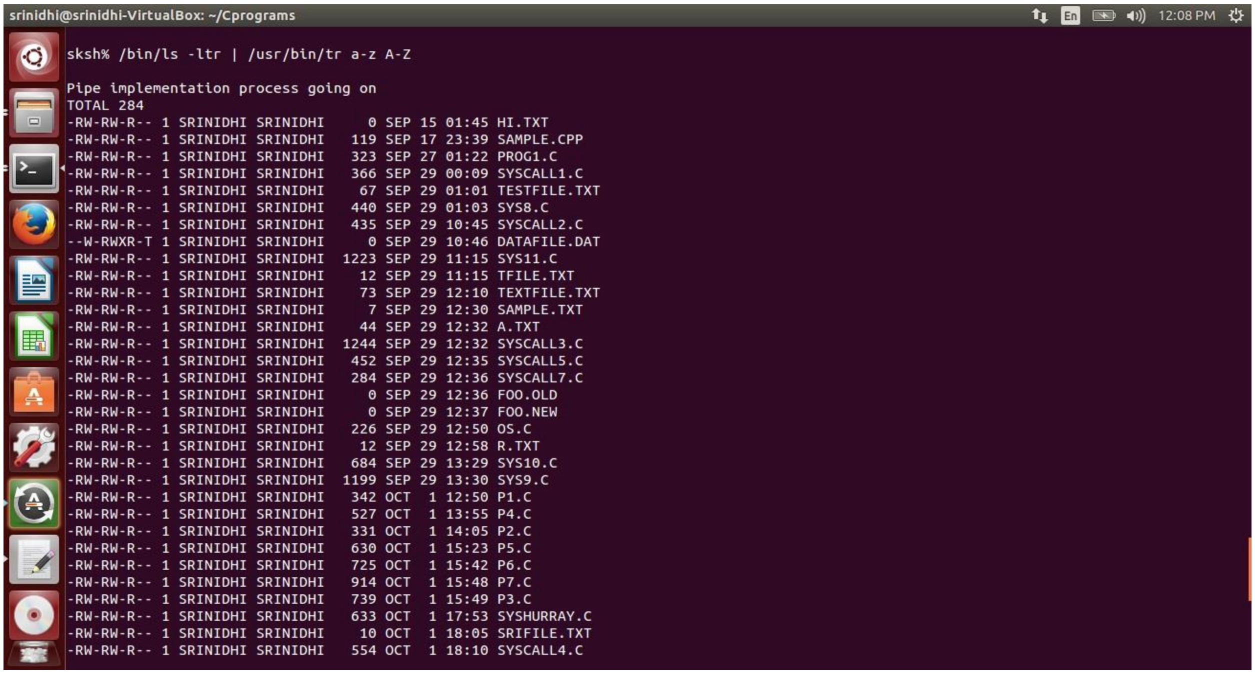
This illustrates the input re direction execution. The input to the command is obtained from the file.



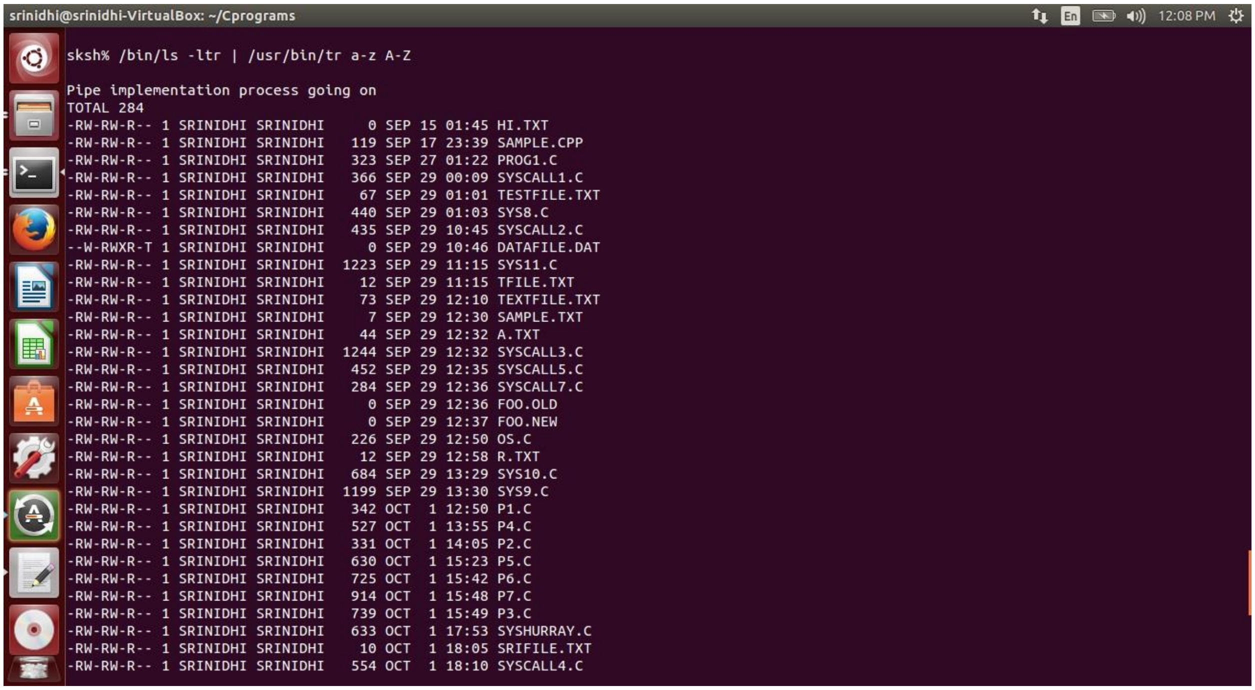
```
-RW-RW-R-- 1 SRINIDHI SRINIDHI   453 OCT  3 11:43 SS1.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI  1456 OCT  3 13:27 SS2.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI  4149 OCT  3 17:05 FINAL.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI  1533 OCT  3 19:32 P8.C
-RW-RW-R-- 1 SRINIDHI SRINIDHI   442 OCT  4 12:36 B1.C
-RW-RW-R-- 1 SRINIDHI 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 SAMLEFILE.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 illustrates the Pipe between two processes. The output of the first process serves as the input of the second process.



```
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 23:39 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
```



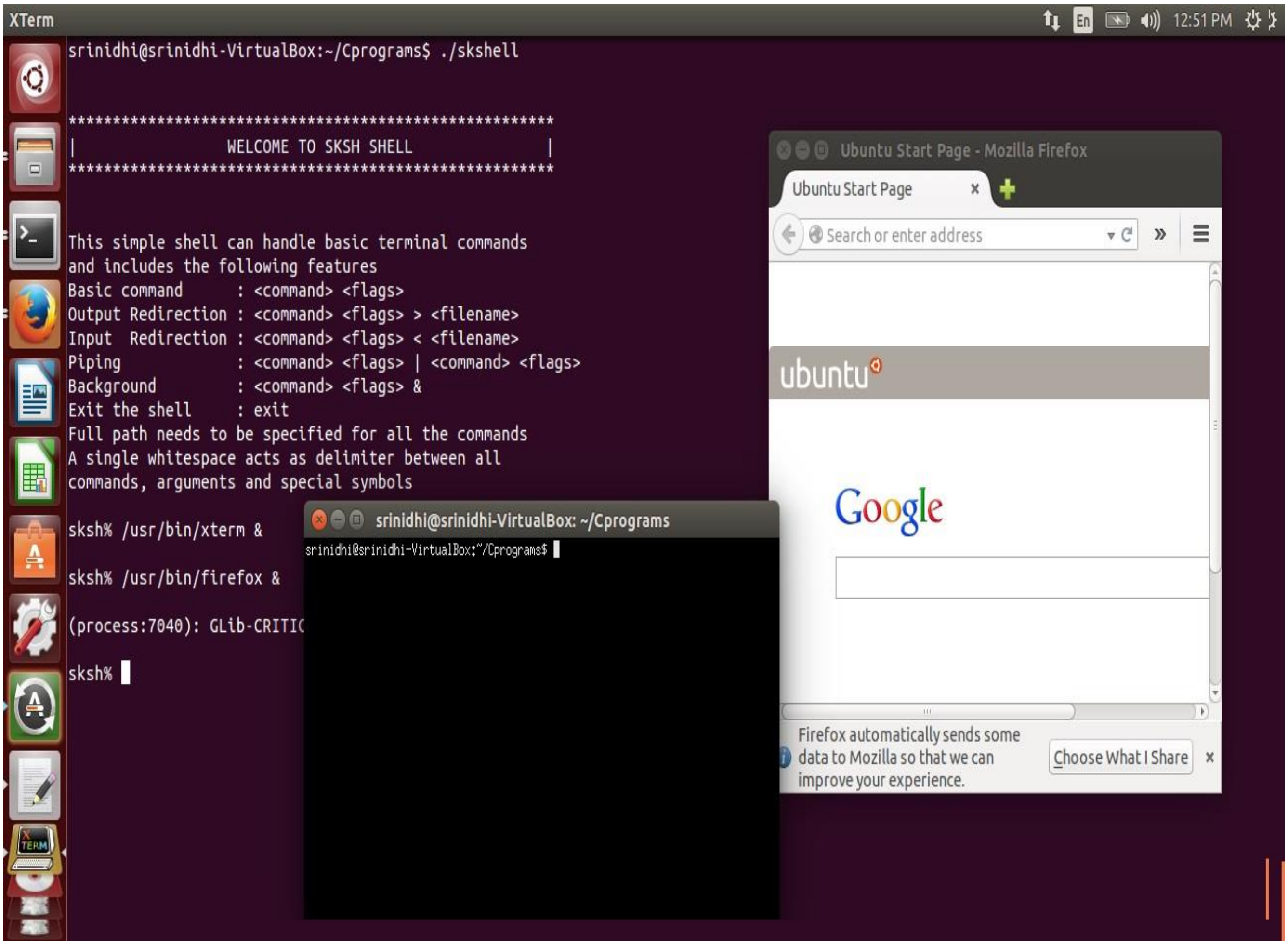
```
sksh% /bin/ls -ltr | /usr/bin/tr a-z A-Z

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 23:39 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
```

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

   /usr/bin/firefox &

This illustrates the execution of background process.

This shell provides the feature of running two background processes.



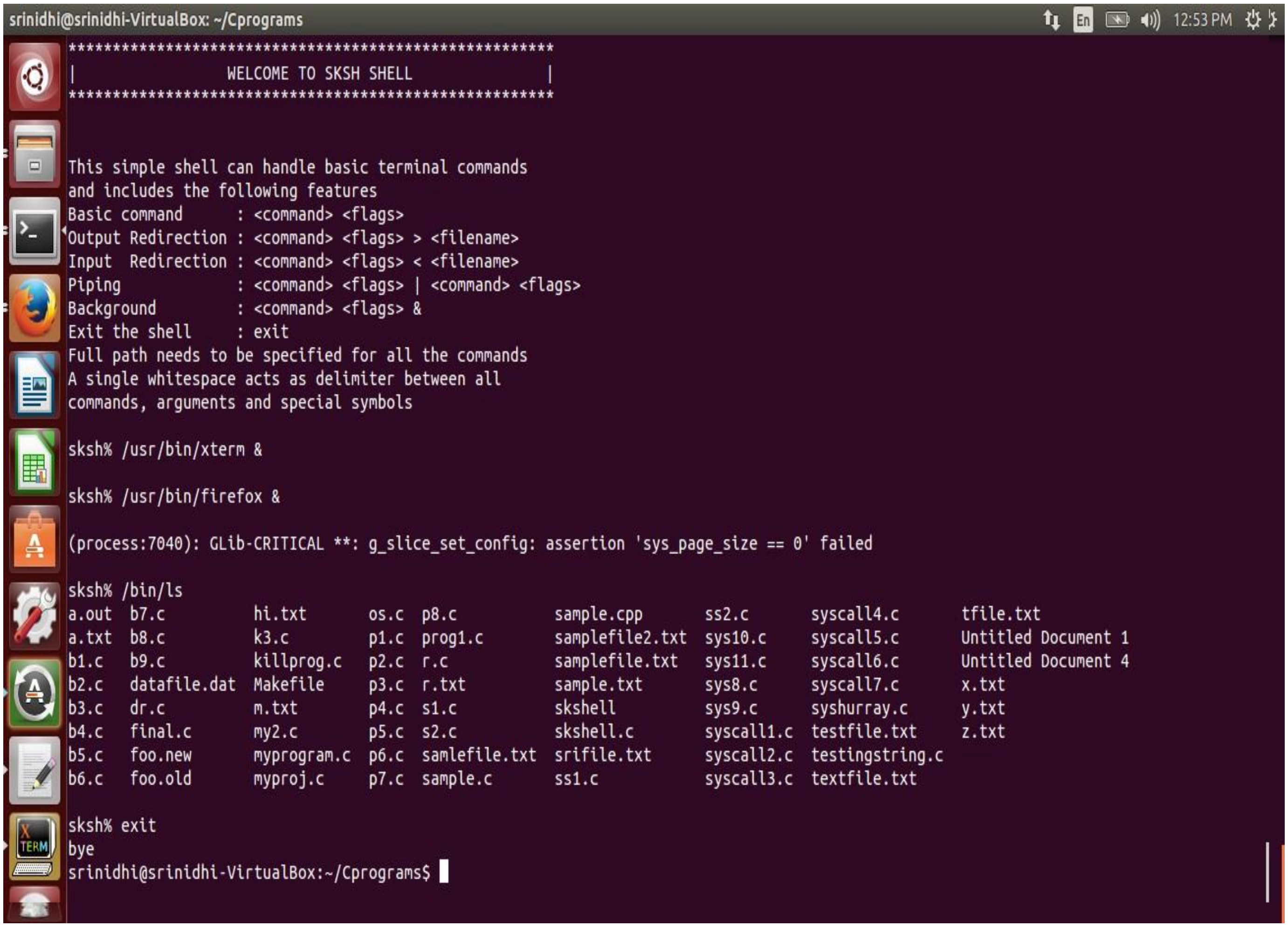
```
srinidhi@srinidhi-VirtualBox:~/Cprograms$ ./skshell


*******************************************************
|                  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/xterm &

sksh% /usr/bin/firefox &

(process:7040): GLib-CRITIC

sksh% 
```

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

Firefox automatically sends some data to Mozilla so that we can improve your experience.

7. Exit shell eg. Exit

This implements the exit from the shell.



```
*******************************************************
|                 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/xterm &

sksh% /usr/bin/firefox &

(process:7040): GLib-CRITICAL **: g_slice_set_config: assertion 'sys_page_size == 0' failed

sksh% /bin/ls
a.out  b7.c          hi.txt       os.c  p8.c           sample.cpp       ss2.c       syscall4.c       tfile.txt
a.txt  b8.c          k3.c         p1.c  prog1.c        samplefile2.txt  sys10.c     syscall5.c       Untitled Document 1
b1.c   b9.c          killprog.c   p2.c  r.c            samplefile.txt   sys11.c     syscall6.c       Untitled Document 4
b2.c   datafile.dat  Makefile     p3.c  r.txt          sample.txt       sys8.c      syscall7.c       x.txt
b3.c   dr.c          m.txt        p4.c  s1.c           skshell          sys9.c      syshurray.c      y.txt
b4.c   final.c       my2.c        p5.c  s2.c           skshell.c        syscall1.c  testfile.txt     z.txt
b5.c   foo.new       myprogram.c  p6.c  samlefile.txt  srifile.txt      syscall2.c  testingstring.c
b6.c   foo.old       myproj.c     p7.c  sample.c       ss1.c            syscall3.c  textfile.txt

sksh% exit
bye
srinidhi@srinidhi-VirtualBox:~/Cprograms$
```