

Assignment 5

OPERATING SYSTEMS

Habiba Gamal

900151007

Introduction:

In this project, I created a C program that acts as a shell interface through which the user inputs commands and they get executed as separate processes.

This was done in several steps.

1) Parsing:

The first step in this project is taking input from stdin and then parsing it into the elements of the args array. This was done through the strtok function that returns a pointer to a string separated by a specified delimiter.

2) Executing command in a child process:

The next challenge was executing the command in a child process. I did this by creating a child through a fork function. Then I check if pid < 0 which means that the fork failed or if the child was created and I am in the child process (pid = 0). This is where execvp was called. Then lastly, I checked if pid > 0 which means that it is the parent process in which the parent waits for the child to finish executing.

3) Supporting '&':

Then I added the support for concurrent execution of the parent and the child. This is done through checking the last element of the args array, if it is '&', a flag is unset and this element in args is changed to NULL to allow for the execution of the command using execvp without errors.

I also changed the condition of the parent waiting to not only just being a parent process. But the waiting is exclusive to parents who are not running concurrently with the child (flag is set).

4) Creating history feature:

The next challenge was creating the history feature. This was done through creating another string called buffer with the same specifications as s (the string that gets inputted). If the inputted command is not "!!", then s is stored in buffer to create history. If the inputted command is "!!", then buffer is assigned to s and the command is outputted to the user then executed.

5) Redirecting input and output:

I implemented this, by looping in the child process through args array and checking if the redirection sign is there. If it is there, I place NULL in its place to avoid errors when executing execvp. Also, I create a file descriptor and then use dup2 to redirect stdin or stdout to a file. Validation is performed in this part to avoid having the redirection symbol as the last character in the line or having 2 redirection symbols following each other. A syntax error is generated in these cases.

6) Communication via a pipe:

This was the most challenging part for me. However, after reading its section in the book it became clear to me. How I did this part, is again by looping in the child through args to check for the pipe symbol. If it is present, I would create another array called args2. Args2 will be the one executed by the grandchild and args will be the one executed by the child. The grandchild will execute the command before the pipe symbol and the child will execute the command after the pipe signal. This is because the command after the pipe needs to wait for the command before the pipe, as the second command uses the output of the first command as input, and the child can wait for the grandchild, but not the other way around. A file descriptor is created, and a pipe is created. For the grandchild, the read end of the pipe is closed

as it won't be used, and the grandchild redirects the output from stdout to the write end of the pipe and then it is closed. As for the child, the writing end of the pipe will be closed as it won't be used, and the reading will be redirected from stdin to the reading end of the pipe then the reading end of the pipe is closed.

Assumptions:

- The same command will not contain both input redirection and output redirection. It will contain either one of them or none at all.
- There can only be 1 pipe character in a command
- Commands containing pipe will not be combined with any redirection operation.

Limitations:

- If the command is ended by a pipe character, the user is not prompted to enter the next pipe command.
- The shell does not support the up and down arrows to get back to past commands.
- The shell does not support command like cd.

How to Run Code:

Cd the directory that includes the project, then run the following two commands to compile and build it from the terminal.

```
gcc -o simple-shell simple-shell.c  
./simple-shell
```

Testcases:

The terminal window shows the following sequence of commands:

```
osc@ubuntu: ~/final-src-osc10e/ch3  
osc@ubuntu: ~$ cd /home/osc/final-src-osc10e/ch3  
osc@ubuntu:~/final-src-osc10e/ch3$ gcc -o simple-shell simple-shell.c  
osc@ubuntu:~/final-src-osc10e/ch3$ ./simple-shell  
osh>ps tree
```

Below this, there is a large screenshot of the ps tree command output, which shows a detailed hierarchy of processes running on the system. The tree starts with lxcfs and includes many system daemons like polkitd, rsyslogd, rtkit-daemon, snapd, sshd, systemd, and upowerd, along with their respective child processes.

```

osc@ubuntu: ~/final-src-osc10e/ch3
|-upowerd-[gdbus]
|  \whoopsie-[gmain]
osh>!!
pstree
systemd-- NetworkManager-[gdbus]
|-accounts-daemon-[gdbus]
|-acpid
|-agetty
|-atd
|-avahi-daemon---avahi-daemon
|-colord-[gdbus]
|-cron
|-cups-browsed-[gdbus]
|-cupsd---[dbus]
|-dbus-daemon
|-dhclient
|-fwupd-[GusbEventThread]
osc@ubuntu: ~/final-src-osc10e/ch3
|  \{main}
|  |  \{probing-thread}
|  \upowerd-[gdbus]
|  \ whoopsie-[gdbus]
|  \ {main}
osh>ls -l | grep Mar
-rw-r--r-- 1 osc osc 0 Mar 31 17:33 <
-rw-r--r-- 1 osc osc 330 Mar 31 09:46 >
-->----- 1 osc osc 14343 Mar 31 11:22 -ael
-rw-r--r-- 1 osc osc 13 Mar 31 17:01 in.txt
-rw-r--r-- 1 osc osc 0 Mar 31 16:04 -l
-rw-r--r-- 1 osc osc 14344 Mar 31 11:26 out10.txt
-->-----x 1 osc osc 0 Mar 31 04:44 out1.txt
-rw-r--r-- 1 osc osc 0 Mar 31 04:52 out2.txt
-rw-r--r-- 1 osc osc 0 Mar 31 04:54 out3.txt
-->----- 1 osc osc 344 Mar 31 11:24 out9.txt
-rw-r--r-- 1 osc osc 14265 Mar 31 11:35 outagain.txt
-rw-r--r-- 1 osc osc 13882 Mar 31 11:36 outconc.txt
-rw-r--r-- 1 osc osc 0 Mar 31 04:56 outg.txt
-rw-r--r-- 1 osc osc 13942 Mar 31 11:26 outs.txt
-rw-r--r-- 1 osc osc 0 Mar 31 04:44 outt.txt
-->----- 1 osc osc 0 Mar 31 04:23 out.txt
osh>[REDACTED]
osc@ubuntu: ~/final-src-osc10e/ch3
osc@ubuntu:~/final-src-osc10e/ch3$ gcc -o simple-shell simple-shell.c
osc@ubuntu:~/final-src-osc10e/ch3$ ./simple-shell
osh>sort < in.txt
10
11
20
5
6
osh>
osc@ubuntu: ~/final-src-osc10e/ch3
osc@ubuntu:~/final-src-osc10e/ch3$ gcc -o simple-shell simple-shell.c
osc@ubuntu:~/final-src-osc10e/ch3$ ./simple-shell
osh>sort < in.txt
10
11
20
5
6
osh>sort <
Syntax error
osh>sort < >
Syntax error
osh>sort > <
Syntax error
osh>sort >
Syntax error
osh>ps -ael > outReport.txt
osh>[REDACTED]

```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	5	0	1	0	0	80	0	-	29995	-	?	00:00:02	systemd
1	5	0	2	0	0	80	0	-	0	-	?	00:00:00	kthreadd
1	5	0	3	2	0	80	0	-	0	-	?	00:00:00	ksoftirqd/0
1	5	0	5	2	0	60	-20	-	0	-	?	00:00:00	kworker/0:0H
1	5	0	6	2	0	80	0	-	0	-	?	00:00:00	kworker/u4:0
1	5	0	7	2	0	80	0	-	0	-	?	00:00:00	rcu_sched
1	5	0	8	2	0	80	0	-	0	-	?	00:00:00	rcu_bh
1	5	0	9	2	0	-40	-	-	0	-	?	00:00:00	migration/0
1	5	0	10	2	0	-40	-	-	0	-	?	00:00:00	watchdog/0
5	5	0	11	2	0	-40	-	-	0	-	?	00:00:00	watchdog/1
1	5	0	12	2	0	-40	-	-	0	-	?	00:00:00	migration/1
1	5	0	13	2	0	80	0	-	0	-	?	00:00:00	ksoftirqd/1
1	5	0	15	2	0	60	-20	-	0	-	?	00:00:00	kworker/1:0H
5	5	0	16	2	0	80	0	-	0	-	?	00:00:00	kdtmpfs
1	5	0	17	2	0	60	-20	-	0	-	?	00:00:00	netns
1	5	0	18	2	0	60	-20	-	0	-	?	00:00:00	perf
1	5	0	19	2	0	80	0	-	0	-	?	00:00:00	khungtaskd
1	5	0	20	2	0	60	-20	-	0	-	?	00:00:00	wrteback
1	5	0	21	2	0	85	5	-	0	-	?	00:00:00	ksnd
1	5	0	22	2	0	99	19	-	0	-	?	00:00:00	khugepaged
1	5	0	23	2	0	60	-20	-	0	-	?	00:00:00	crypto
1	5	0	24	2	0	60	-20	-	0	-	?	00:00:00	kintegrityd
1	5	0	25	2	0	60	-20	-	0	-	?	00:00:00	bioset
1	5	0	26	2	0	60	-20	-	0	-	?	00:00:00	kblockd
1	5	0	27	2	0	60	-20	-	0	-	?	00:00:00	ata_sff
1	5	0	28	2	0	60	-20	-	0	-	?	00:00:00	md
1	5	0	29	2	0	60	-20	-	0	-	?	00:00:00	devfreq_wq
1	5	0	30	2	0	80	0	-	0	-	?	00:00:00	kworker/u4:1
1	5	0	33	2	0	80	0	-	0	-	?	00:00:00	kswapd0
1	5	0	34	2	0	60	-20	-	0	-	?	00:00:00	vnstat
1	5	0	35	2	0	80	0	-	0	-	?	00:00:00	fsnotify_mark
1	5	0	36	2	0	80	0	-	0	-	?	00:00:00	cryptofs-kthrea
1	5	0	52	2	0	60	-20	-	0	-	?	00:00:00	kthrotld
1	5	0	53	2	0	60	-20	-	0	-	?	00:00:00	acpl_thermal_pm
1	5	0	54	2	0	60	-20	-	0	-	?	00:00:00	bioset
1	5	0	55	2	0	60	-20	-	0	-	?	00:00:00	bioset
1	5	0	56	2	0	60	-20	-	0	-	?	00:00:00	bioset
1	5	0	57	2	0	60	-20	-	0	-	?	00:00:00	bioset
1	5	0	58	2	0	60	-20	-	0	-	?	00:00:00	bioset
1	5	0	59	2	0	60	-20	-	0	-	?	00:00:00	bioset
1	5	0	60	2	0	60	-20	-	0	-	?	00:00:00	bioset

```
osh@ubuntu:~/final-src-osc10e/ch3
osh>ps -ael > outReport.txt
osh>ls &
osh><
      fig3-35.c      out3.txt      out.txt
>     in.txt       out9.txt      pid.c
-ael      -l      outagain.txt  shm-posix-consumer.c
DateClient.java  multi-fork      outconc.txt  shm-posix-producer.c
DateServer.java  multi-fork.c    outg.txt    simple-shell.c
fig3-30.c        newproc-postx.c outReport.txt  simple-shell.c
fig3-31.c        newproc-wln32.c outst.txt   unix_pipe.c
fig3-32.c        out10.txt     outtree2.txt  win32-pipe-child.c
fig3-33.c        out1.txt      outtree.txt   win32-pipe-parent.c
fig3-34.c        out2.txt      outtt.txt
ls -l
total 244
-rw-r--r-- 1 osc osc 0 Mar 31 17:33 <
-rw-r--r-- 1 osc osc 330 Mar 31 09:46 >
----- 1 osc osc 1443 Mar 31 11:22 -ael
-rw-rw-r- 1 osc osc 710 Jan 3 2018 DateClient.java
-rw-rw-r- 1 osc osc 840 Jun 18 2018 DateServer.java
-rw-rw-r- 1 osc osc 361 Jun 18 2018 fig3-30.c
-rw-rw-r- 1 osc osc 121 Jan 3 2018 fig3-31.c
-rw-rw-r- 1 osc osc 136 Jan 3 2018 fig3-32.c
-rw-rw-r- 1 osc osc 500 Jun 18 2018 fig3-33.c
-rw-rw-r- 1 osc osc 680 Jun 18 2018 fig3-34.c
```

Input of spaces and tabs:

```
osh>ps tree
systemd--NetworkManager---{gdbus}
|   accounts-daemon---{gdbus}
|   acpid
|   agetty
|   atd
|   avahi-daemon---avahi-daemon
|   color---{gdbus}
|   cron
|   cups-browsed---{gdbus}
|   cupsd---9*[dbus]
|   dbus-daemon
```

Moreover, all the testcases provided today by the teaching assistant were tested and worked successfully.