PROJECT NAME: CS 415 Programming Assignment #1

Test Priority (LOW/MED./HIGH): HIGH	**Test Designed By:** Adam Moses, Andrew Jordan
Module Name: Simple command-line	
Test Title: Command-Line Test Suite	
Description: Test the simple command-line interpreter	
Caveats: Linux environment (req.) , Code Blocks IDE (Linux) and Eclipse IDE (Linux) where used for developing and testing	

Test Case	Test Data	Expected Results	Actual results	Status (Pass / Fail)
Prompt a command	>>	>>	>>	P
Read a command	echo here is some text to a file > sample1.txt	Command read from shell, parsed, executed, and output redirected to file	Command read, parsed, executed, and output redirected to file	p
Parse a command	echo here is some text to a file > sample1.txt	echo | here | is | some |text| to| a| file| >| sample1.txt|	echo | here | is | some |text| to| a| file| >| sample1.txt|	p
Correctly respond to signals (break, EOF, etc.)	/r Ctrl-C	Send string to be parsed Signal caught and then kill and exit process	Sent string to be parsed Killed and exited process	P p
Exercise the *cd* and *pwd* command	cd ../.. cd .. pwd cd / cd	Move up 2 dir Move up 1 dir Display cwd Move up to root Display cwd	Moved up 2 dir Moved up 1 dir Displayed cwd Moved up to root Display cwd	P P P P p

Exercise the redirect I/O	ls -l > lstestfile.txt	Send output to file	Sent output to file given	P
	echo here is some text to a file > sample1.txt			
	echo here is some appended text to my file >> sample1.tst	Append text to file	Sent output to be appended to file	P
	less -FX sample1.txt	Display output from file	Displayed output from file	P
	ls < LStest1.txt	Read in command and exe	Read in -l and executed ls -l	P
	echo < sample1.txt	Read in text and echoed to screen	Read text and outputted to screen	P

NOTES:

```
/*
 * main.cpp
 *
 *  Created on: Jan 26, 2017
 *      Author: Andrew Jordan
 *                  Lucas Pruitt
 *                  Adam Moses
 *
 * Purpose: Simulate bash shell
 * Design:
 * If input from shell calling program
 * - copy arguments,fork,execute commands
 * If no input from shell calling program:
 * - Grab line of input
 * - Store in cstring
 * - Parse cstring
 *            - convert each individual command to string
 *            - use length of string to allocate new memory for char[]
 *            - put in Q of commands converted
 *            - use size of Q to allocate new memory for struct char[]
 *            - transfer all commands to the struct char[] from Q
 *            - add any > || >> || < || <<
```

```
 *      - Check if directory change command
 *      - If not changing directories
 *              - fork
 *              - have child check out I/O redirect and do so
 *              - have child process execute command
 *              - kill child process off
 *              - return to parent process
 *      - If exit or ctrl-C pressed, exit process
 *      - Repeat all steps above in infinite while in main.
 */

#include <iostream>
#include <unistd.h>
// used with chdir() -> changes working directory
#include <sys/wait.h>
#include <sys/types.h>
#include <stdlib.h>
#include <dirent.h>
#include <string>
#include <ctype.h>
#include <algorithm>
#include <string.h>
#include <stdio.h>
#include <signal.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <queue>
#include <signal.h>
using namespace std;


struct args
{
        int argc;
        char **argv;
};

args* parser(char*);  // Parses arguments and stores them inside an args struct
void order(char*);      // Sets up order of functions called for reuse if needed
char* S2C(string,int);// Converts a string to a cstring, returns a char* to be stored in args
bool dir(args*);        // Changes and displays directories
void USER_PWD();        // Displays current working directory
void fork_off(args *);// Creates child process to do work and child is terminated with kill(pid,SIGTERM);
void sig_handler (int);// Catches ctrl-C and calls quit_process()
```

```
void quit_process();  // Terminates current process with kill(pid,SIGTERM)


int main(int argc,char *argv[])
{
	if (argc>1)
	{
		args *aptr = new args;
		aptr->argc = argc;
		aptr->argv = new char*[argc+1];
		for (int i=1;i<argc;i++)
			aptr->argv[i-1]=argv[i];
		aptr->argv[aptr->argc] = '\0';
		fork_off(aptr);
	}

	char input[1500];
	while (true)
	{
		if (signal(SIGINT,sig_handler)== SIG_ERR)
		{
			cout<<"Signal not caught"<<endl;
		}
		for (int i=0;i<1500;i++)
				input[i]= '\0';


		fgets(input,1500,stdin);
		order(input);
	}// end infinite while
	return 0;
}
void order(char *input)
{
	args *aptr;
	aptr = parser(input);
	fork_off(aptr);
}
void fork_off(args *aptr)
{
	bool runexec;
	int checkEXEC;
	if (aptr->argv[0]== std::string("exit_"))
		quit_process();
```

```
runexec = dir(aptr);
        if (!runexec)
        {
                pid_t REpid = fork(); // returned pid
                                switch (REpid)
                                {
                                        case -1:
                                                perror ("fork");
                                                exit(1);
                                                break;
                                        case 0:
                                        {
                                                cout<<">>Child process:"<<REpid<<" starting up.."<<endl;

                                                int m = 0;
                                                for (int i = 0; i < aptr->argc; i++)
                                                  {
                                                    if (aptr->argv[i] == std::string(">") ||
                                                        aptr->argv[i] == std::string("<") ||
                                                        aptr->argv[i] == std::string("<<") ||
                                                        aptr->argv[i] == std::string(">>"))
                                                      {m = i; }

                                                  }
                                                if (aptr->argv[m] == std::string(">"))
                                                  {

                                                                int newfd = open(aptr->argv[m + 1],O_CREAT
                                                                        |O_WRONLY|O_TRUNC, 0644);
                                                        close(STDOUT_FILENO);
                                                        dup2(newfd, 1);
                                                        aptr->argv[m] = NULL;
                                                        checkEXEC = execvp(aptr->argv[0], aptr->argv);
                                                  }
                                                if (aptr->argv[m] == std::string(">>"))
                                                  {
                                                        int newfd = open(aptr->argv[m + 1],
                                                                        O_CREAT|O_WRONLY|O_APPEND, 0644);
                                                        close(STDOUT_FILENO);
                                                        dup2(newfd, 1);
                                                        aptr->argv[m] = NULL;
                                                        checkEXEC = execvp(aptr->argv[0], aptr->argv);
                                                  }
```

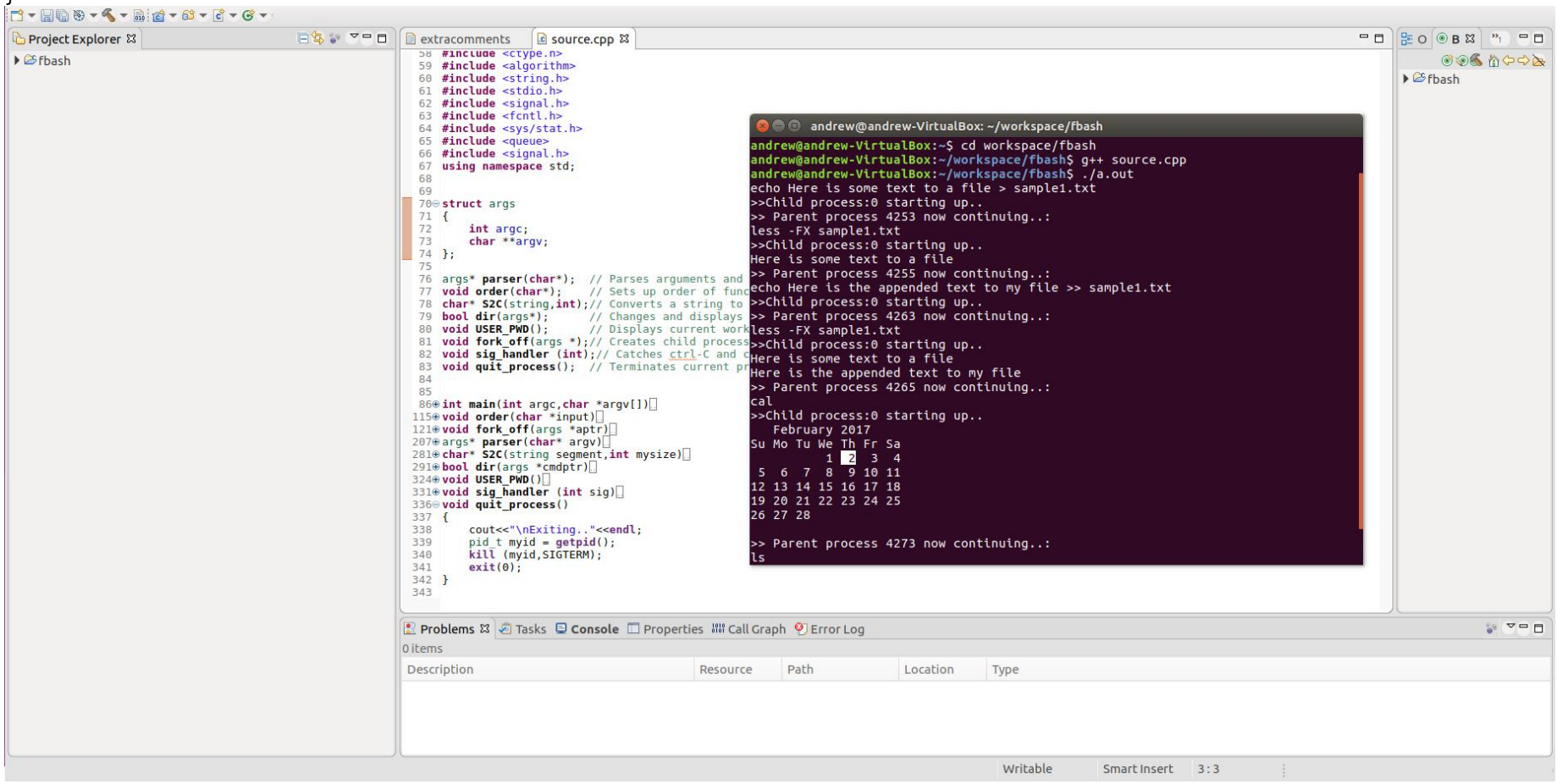
```
                                        if (aptr->argv[m] == std::string("<"))
                                          {
                                                  char buffer[1000];
                                                  for(int i = 0; i < 1000; i++)
                                                          buffer[i]= '\0';
                                                auto newID = open(aptr->argv[m+1], O_CREAT
                                                  |O_RDONLY, 0644);

                                                if(read(newID, buffer, 1000) == -1)
                                                  exit(-1);

                                                // Parse commands from file and update args
                                                args *temp = parser(buffer);
                                                args *newarguments = new args;
                                                newarguments->argv = new char* [(temp->argc)+2];
                                                  newarguments->argv[0] = aptr->argv[m -1];
                                                for (int i=1;i< (temp->argc)+1;i++)
                                                  newarguments->argv[i] = temp->argv[i-1];

                                                newarguments->argv[(temp->argc)+1] = '\0';
                                                //execute new args
                                                execvp(newarguments->argv[0], newarguments->argv);
                                          }

                                        else
                                         checkEXEC = execvp (aptr->argv[0], aptr->argv);

                                        if (checkEXEC == -1)
                                                perror("exec");
                                        break;
                                }
                                default:
                                        if (wait(0)==-1)
                                                perror("wait");
                                        cout<<">> Parent process "<<REpid<<" now continuing..:"<<endl;
                                        break;
                        }//end switch
                cout<<"Child process now dieing.."<<endl;
                kill (REpid,SIGTERM);
                }// if runexec
}// end fork off
args* parser(char* argv)
{
        args *arguments = new args;
```

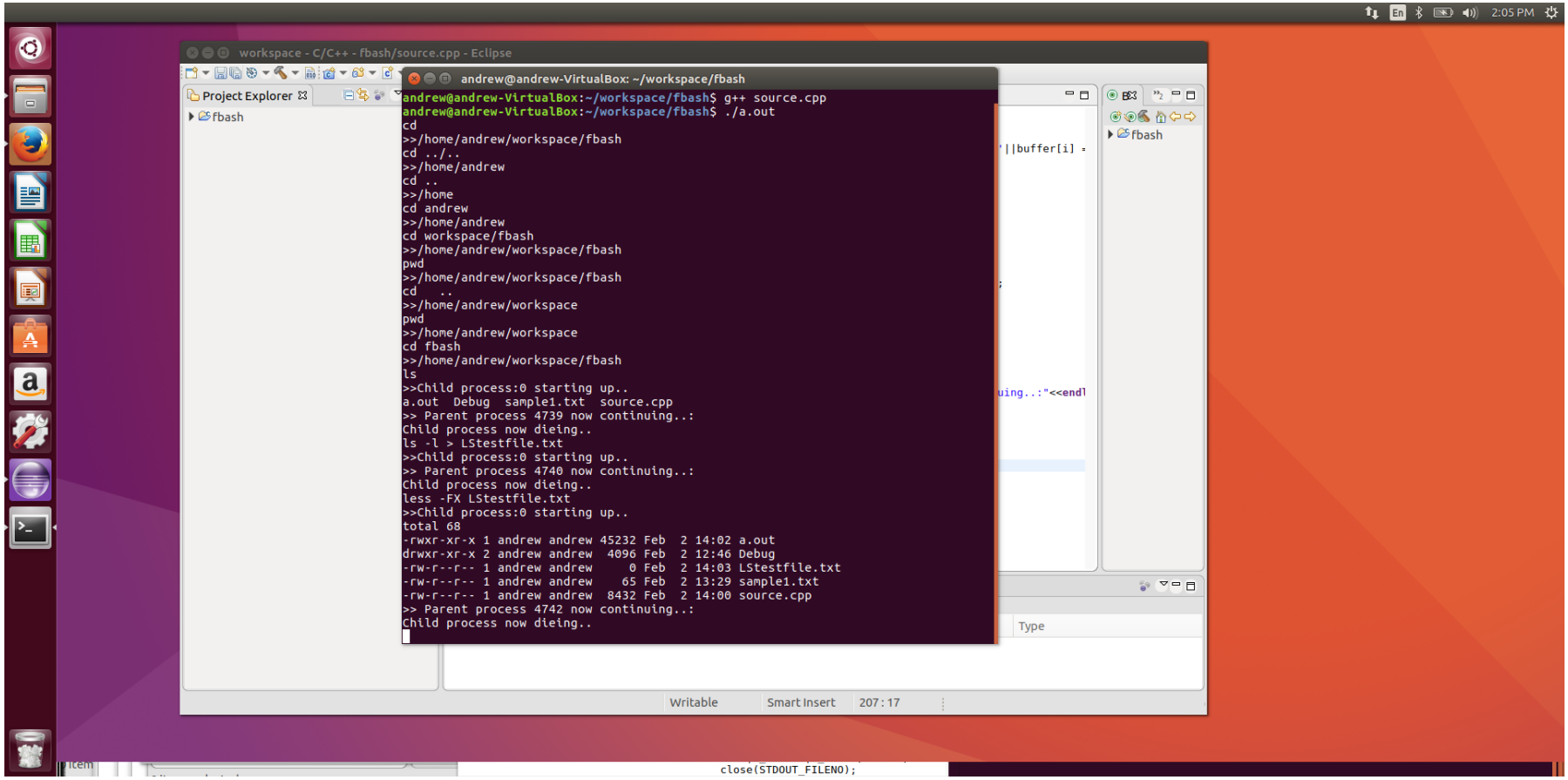
```
string segment = "";
int segment_size = 0;
int i = 0;
char *GGRTR = new char[3]{'>', '>', '\0'};
char *LLESS = new char[3]{'<', '<', '\0'};
char *GRTR = new char[2]{'>', '\0'};
char *LESS = new char[2]{'<', '\0'};
bool store_command = false;
queue<char*> Qcommands;

while (argv[i] != '\0')
{
	while ((argv[i] != ' ')
		&& (argv[i] != '\n')
		&& (argv[i] != '\r')
		&& (argv[i] != '\t')
		&& (argv[i] != '>')
		&& (argv[i] != '<'))
	{
		segment += argv[i];
		segment_size++;
		i++;
		store_command = true;
	}
	if (store_command)
	{
		Qcommands.push(S2C(segment, segment_size));
		segment = "";
		segment_size = 0;
		store_command = false;
	}

	if (argv[i] == '>')
	{
		int double_check = i;
		if (argv[double_check + 1] == '>')
		{
			Qcommands.push(GGRTR);
			i++;
		}
		if (argv[double_check + 1] != '>')
			Qcommands.push(GRTR);
	}//end if >
```

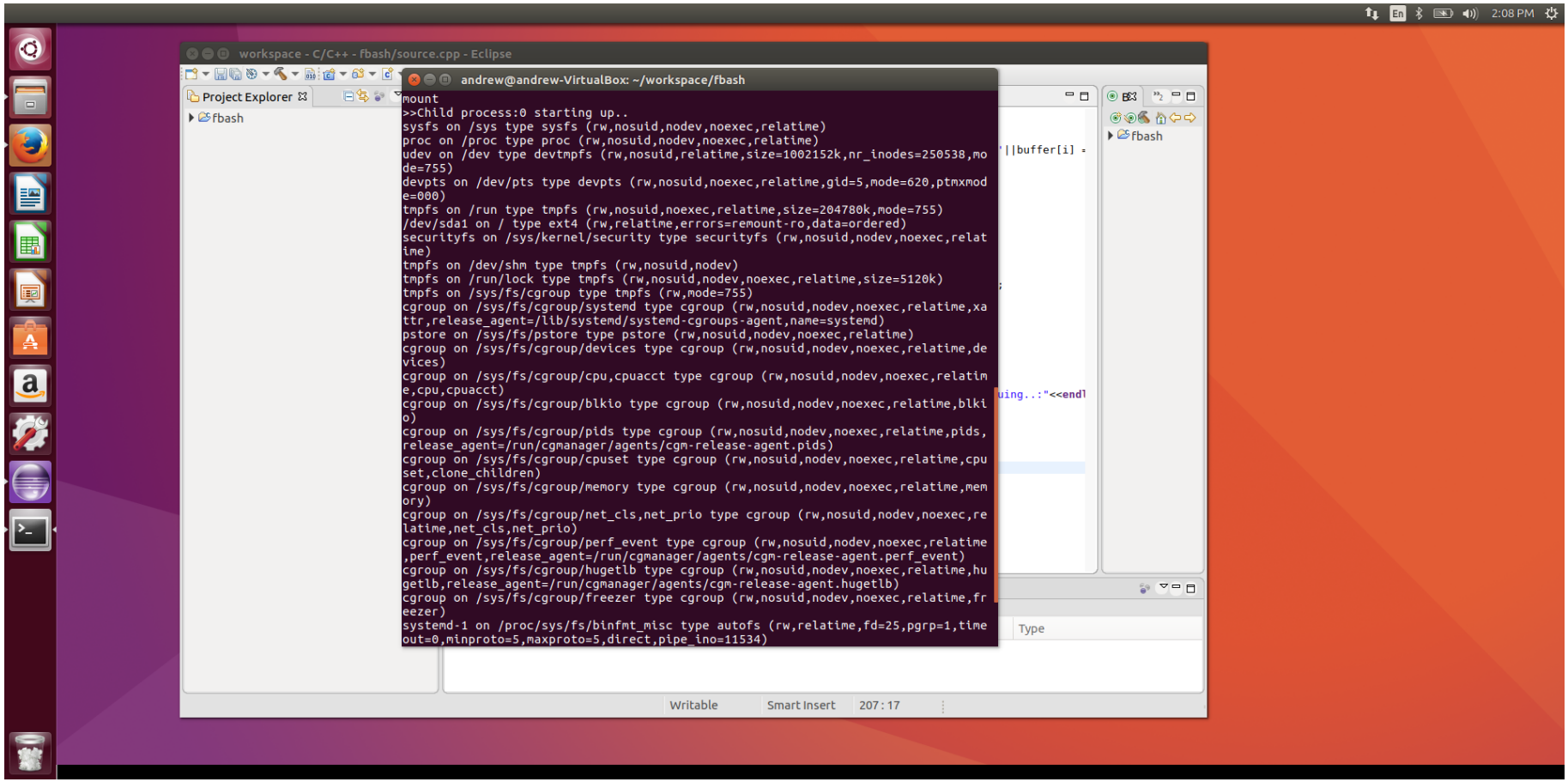
```
                if (argv[i] == '<')
                {
                        int double_check = i;
                        if (argv[double_check + 1] == '<')
                        {
                                Qcommands.push(LLESS);
                                i++;
                        }
                        if (argv[double_check + 1] != '<')
                                Qcommands.push(LESS);
                }//end if <
                i++;
        }//end while

        //Allocate memory for arguments inside of struct
        int counter = Qcommands.size();
        arguments->argc = counter;
        arguments->argv = new char*[counter+1];

        // Transfer commands into struct
        for (int i = 0; i < counter; i++,Qcommands.pop())
                arguments->argv[i] = Qcommands.front();

        arguments->argv[counter] = '\0';
        //return pointer containing arguments
        return arguments;
}
char* S2C(string segment,int mysize)
{
        //grab new memory for cstring
        char *temp = new char[mysize+1];
        // transfer characters from string to new char array
        for (int i = 0; i < mysize; i++)
                temp[i] = segment[i];
        temp[mysize] = '\0';
        return temp;
}
bool dir(args *cmdptr)
{
        int changedir_test;
        bool temp = false;
        // cycle through commands
        for (int i =0;i<cmdptr->argc;i++)
        {
```

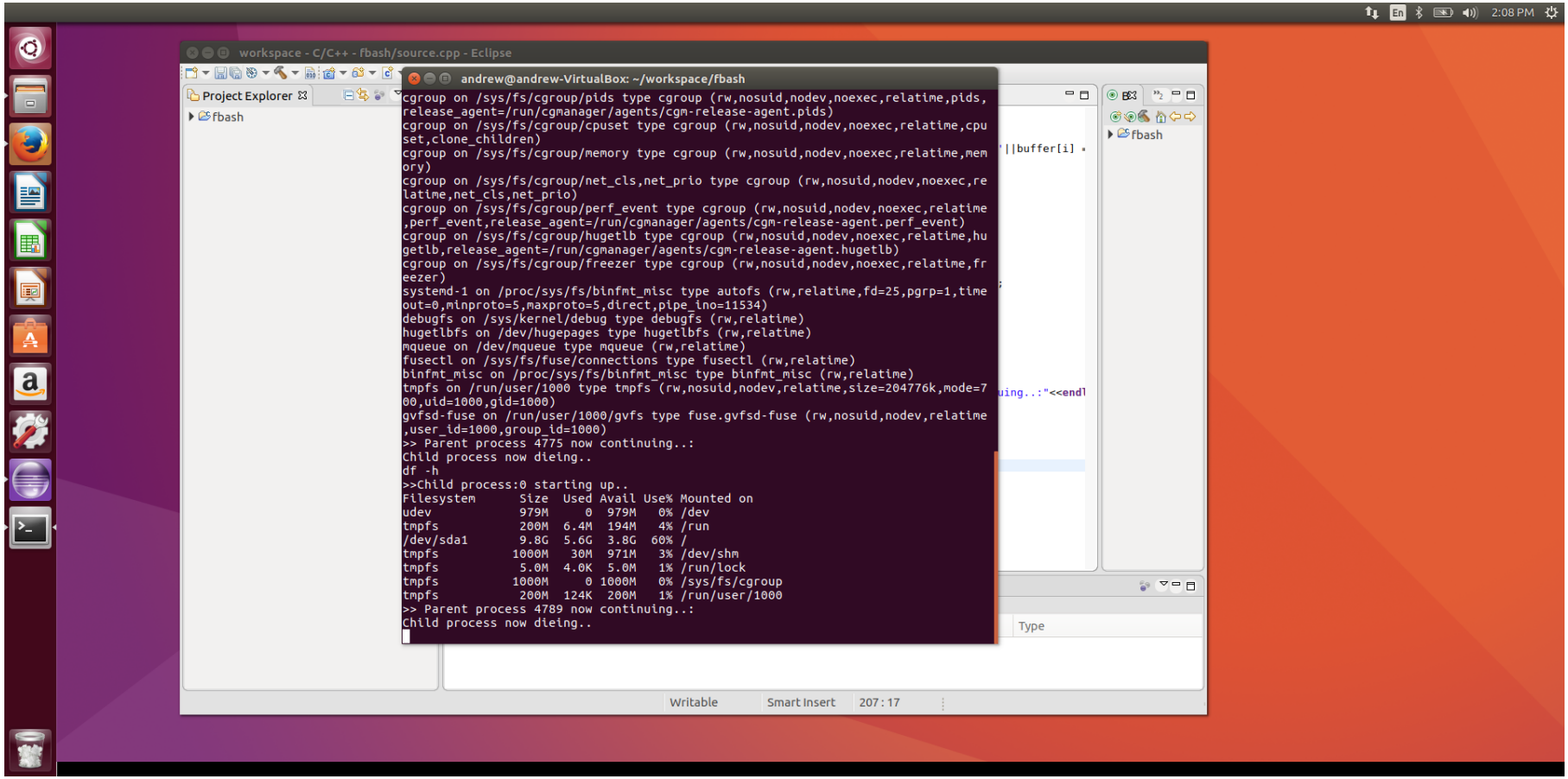
```
		if (cmdptr->argv[i]==std::string("cd"))
		{
			if (cmdptr->argv[i+1]=='\0')
				changedir_test = 1;
			else if (cmdptr->argv[i+1]==std::string(".."))
				 changedir_test = chdir("..");
			else if (cmdptr->argv[i+1]==std::string("../.."))
				 changedir_test = chdir("../..");
			else if (cmdptr->argv[i+1]!='\0')
				changedir_test = chdir(cmdptr->argv[i+1]);

			if (changedir_test == -1)
				cout<<">>Error:Did not change directories."<<endl;
			if (changedir_test != -1)
				{
					USER_PWD();
					return true;
				}
		}// if cd

		if (cmdptr->argv[i]==std::string("pwd"))
			{
				USER_PWD();
				return true;
			}
	}// end for
return temp;
}// end non_exe
void USER_PWD()
{
	char buffer[200];
	char *newpath = getcwd(buffer,200);
	string currpath = newpath;
	cout<<">>"<<currpath<<endl;
}
void sig_handler (int sig)
{
	if (sig == SIGINT)
		quit_process();
}
void quit_process()
{
	cout<<"\nExiting.."<<endl;
	pid_t myid = getpid();
```

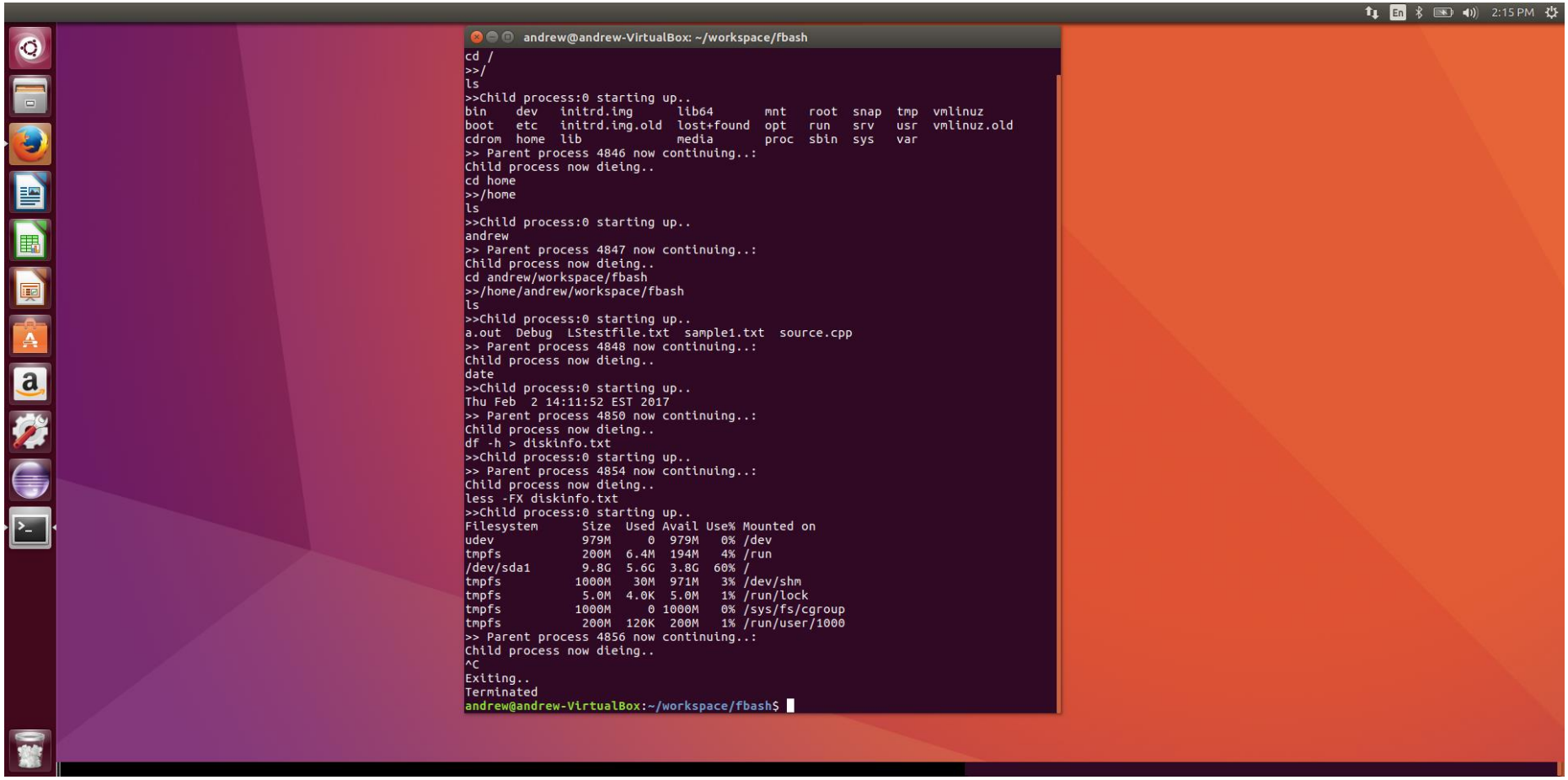
```
        kill (myid,SIGTERM);
        exit(0);
}
```

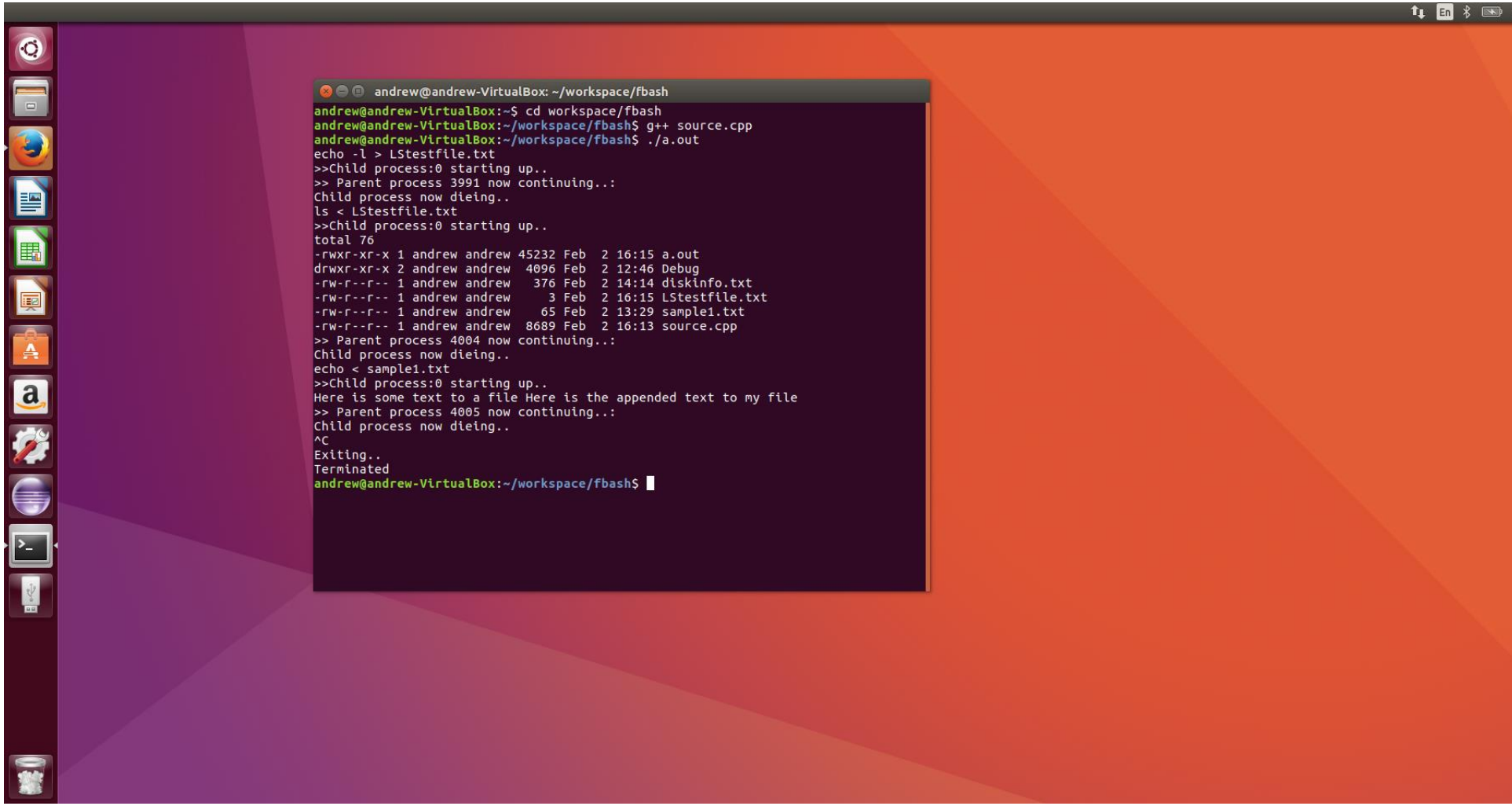



2:05 PM
workspace - C/C++ - fbash/source.cpp - Eclipse
Project Explorer
fbash
andrew@andrew-VirtualBox: ~/workspace/fbash
andrew@andrew-VirtualBox:~/workspace/fbash$ g++ source.cpp
andrew@andrew-VirtualBox:~/workspace/fbash$ ./a.out
cd
>>/home/andrew/workspace/fbash
cd ../..
>>/home/andrew
cd ..
>>/home
cd andrew
>>/home/andrew
cd workspace/fbash
>>/home/andrew/workspace/fbash
pwd
>>/home/andrew/workspace/fbash
cd ..
>>/home/andrew/workspace
pwd
>>/home/andrew/workspace
cd fbash
>>/home/andrew/workspace/fbash
ls
>>Child process:0 starting up..
a.out Debug sample1.txt source.cpp
>> Parent process 4739 now continuing..:
Child process now dieing..
ls -l > LStestfile.txt
>>Child process:0 starting up..
>> Parent process 4740 now continuing..:
Child process now dieing..
less -FX LStestfile.txt
>>Child process:0 starting up..
total 68
-rwxr-xr-x 1 andrew andrew 45232 Feb 2 14:02 a.out
drwxr-xr-x 2 andrew andrew 4096 Feb 2 12:46 Debug
-rw-r--r-- 1 andrew andrew 0 Feb 2 14:03 LStestfile.txt
-rw-r--r-- 1 andrew andrew 65 Feb 2 13:29 sample1.txt
-rw-r--r-- 1 andrew andrew 8432 Feb 2 14:00 source.cpp
>> Parent process 4742 now continuing..:
Child process now dieing..
Type
Writable
Smart Insert
207 : 17
close(STDOUT_FILENO);


2:08 PM
workspace - C/C++ - fbash/source.cpp - Eclipse
Project Explorer
fbash
andrew@andrew-VirtualBox: ~/workspace/fbash
mount
>>Child process:0 starting up..
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1002152k,nr_inodes=250538,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=204780k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (rw,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids,release_agent=/run/cgmanager/agents/cgm-release-agent.pids)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset,clone_children)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event,release_agent=/run/cgmanager/agents/cgm-release-agent.perf_event)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb,release_agent=/run/cgmanager/agents/cgm-release-agent.hugetlb)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=25,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=11534)
Type
Writable
Smart Insert
207 : 17


workspace - C/C++ - fbash/source.cpp - Eclipse
Project Explorer
fbash
andrew@andrew-VirtualBox: ~/workspace/fbash
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids,release_agent=/run/cgmanager/agents/cgm-release-agent.pids)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset,clone_children)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event,release_agent=/run/cgmanager/agents/cgm-release-agent.perf_event)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb,release_agent=/run/cgmanager/agents/cgm-release-agent.hugetlb)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=25,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=11534)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
mqueue on /dev/mqueue type mqueue (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=204776k,mode=700,uid=1000,gid=1000)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
>> Parent process 4775 now continuing..:
Child process now dieing..
df -h
>>Child process:0 starting up..
Filesystem Size Used Avail Use% Mounted on
udev 979M 0 979M 0% /dev
tmpfs 200M 6.4M 194M 4% /run
/dev/sda1 9.8G 5.6G 3.8G 60% /
tmpfs 1000M 30M 971M 3% /dev/shm
tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs 1000M 0 1000M 0% /sys/fs/cgroup
tmpfs 200M 124K 200M 1% /run/user/1000
>> Parent process 4789 now continuing..:
Child process now dieing..
Type
Writable
Smart Insert
207 : 17


2:15 PM
andrew@andrew-VirtualBox: ~/workspace/fbash
cd /
>>/
ls
>>Child process:0 starting up..
bin dev initrd.img lib64 mnt root snap tmp vmlinuz
boot etc initrd.img.old lost+found opt run srv usr vmlinuz.old
cdrom home lib media proc sbin sys var
>> Parent process 4846 now continuing..:
Child process now dieing..
cd home
>>/home
ls
>>Child process:0 starting up..
andrew
>> Parent process 4847 now continuing..:
Child process now dieing..
cd andrew/workspace/fbash
>>/home/andrew/workspace/fbash
ls
>>Child process:0 starting up..
a.out Debug LStestfile.txt sample1.txt source.cpp
>> Parent process 4848 now continuing..:
Child process now dieing..
date
>>Child process:0 starting up..
Thu Feb 2 14:11:52 EST 2017
>> Parent process 4850 now continuing..:
Child process now dieing..
df -h > diskinfo.txt
>>Child process:0 starting up..
>> Parent process 4854 now continuing..:
Child process now dieing..
less -FX diskinfo.txt
>>Child process:0 starting up..
Filesystem Size Used Avail Use% Mounted on
udev 979M 0 979M 0% /dev
tmpfs 200M 6.4M 194M 4% /run
/dev/sda1 9.8G 5.6G 3.8G 60% /
tmpfs 1000M 30M 971M 3% /dev/shm
tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs 1000M 0 1000M 0% /sys/fs/cgroup
tmpfs 200M 120K 200M 1% /run/user/1000
>> Parent process 4856 now continuing..:
Child process now dieing..
^C
Exiting..
Terminated
andrew@andrew-VirtualBox:~/workspace/fbash$


andrew@andrew-VirtualBox: ~/workspace/fbash
andrew@andrew-VirtualBox:~$ cd workspace/fbash
andrew@andrew-VirtualBox:~/workspace/fbash$ g++ source.cpp
andrew@andrew-VirtualBox:~/workspace/fbash$ ./a.out
echo -l > LStestfile.txt
>>Child process:0 starting up..
>> Parent process 3991 now continuing..:
Child process now dieing..
ls < LStestfile.txt
>>Child process:0 starting up..
total 76
-rwxr-xr-x 1 andrew andrew 45232 Feb 2 16:15 a.out
drwxr-xr-x 2 andrew andrew 4096 Feb 2 12:46 Debug
-rw-r--r-- 1 andrew andrew 376 Feb 2 14:14 diskinfo.txt
-rw-r--r-- 1 andrew andrew 3 Feb 2 16:15 LStestfile.txt
-rw-r--r-- 1 andrew andrew 65 Feb 2 13:29 sample1.txt
-rw-r--r-- 1 andrew andrew 8689 Feb 2 16:13 source.cpp
>> Parent process 4004 now continuing..:
Child process now dieing..
echo < sample1.txt
>>Child process:0 starting up..
Here is some text to a file Here is the appended text to my file
>> Parent process 4005 now continuing..:
Child process now dieing..
^C
Exiting..
Terminated
andrew@andrew-VirtualBox:~/workspace/fbash$