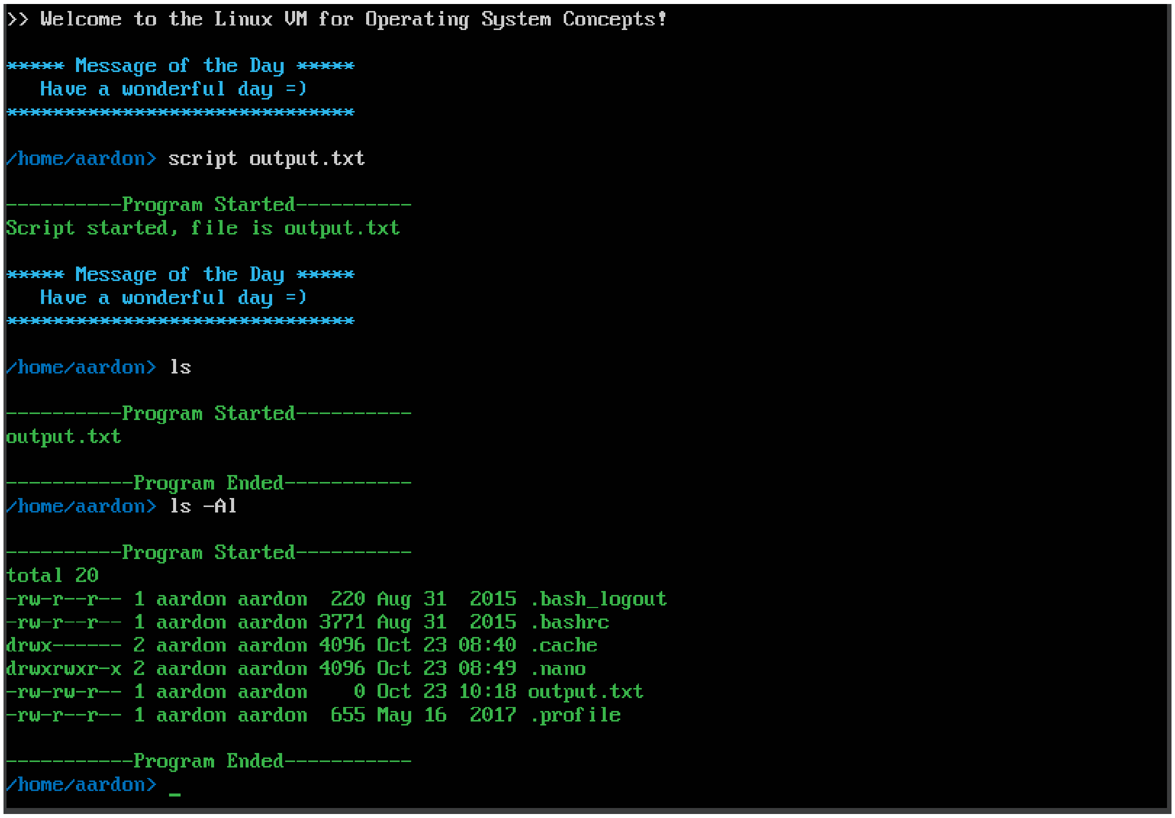
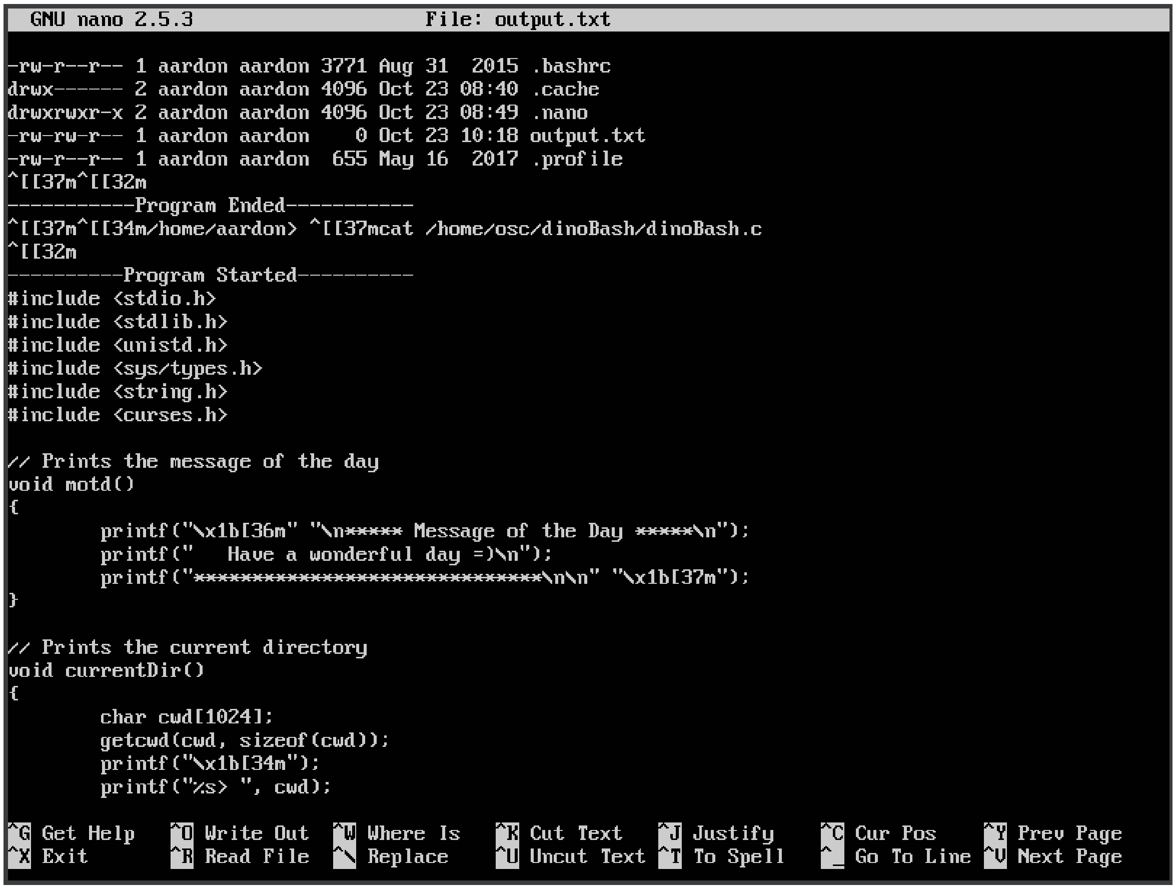
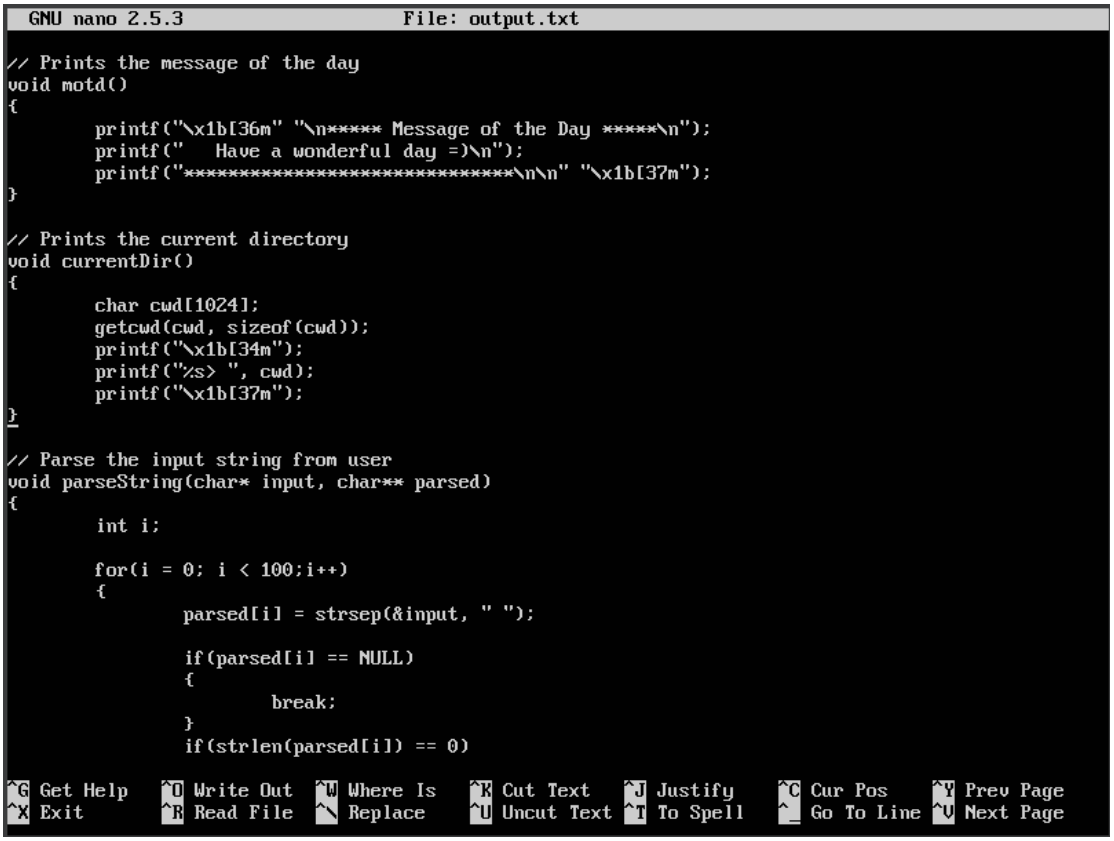


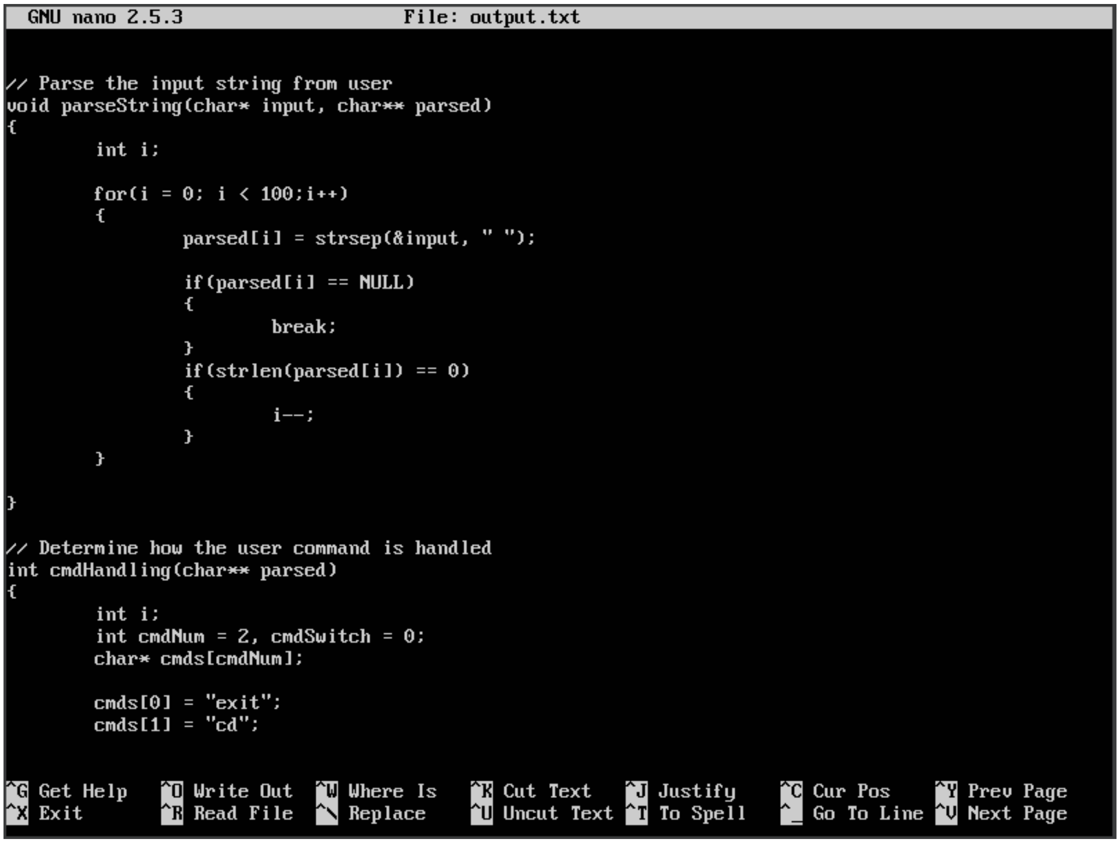
```
>> Welcome to the Linux VM for Operating System Concepts!

***** Message of the Day *****
   Have a wonderful day =)
******************************

/home/aardon> script output.txt

----------Program Started----------
Script started, file is output.txt

***** Message of the Day *****
   Have a wonderful day =)
******************************

/home/aardon> ls

----------Program Started----------
output.txt

-----------Program Ended-----------
/home/aardon> ls -Al

----------Program Started----------
total 20
-rw-r--r-- 1 aardon aardon  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 aardon aardon 3771 Aug 31  2015 .bashrc
drwx------ 2 aardon aardon 4096 Oct 23 08:40 .cache
drwxrwxr-x 2 aardon aardon 4096 Oct 23 08:49 .nano
-rw-rw-r-- 1 aardon aardon    0 Oct 23 10:18 output.txt
-rw-r--r-- 1 aardon aardon  655 May 16  2017 .profile

-----------Program Ended-----------
/home/aardon> _
```



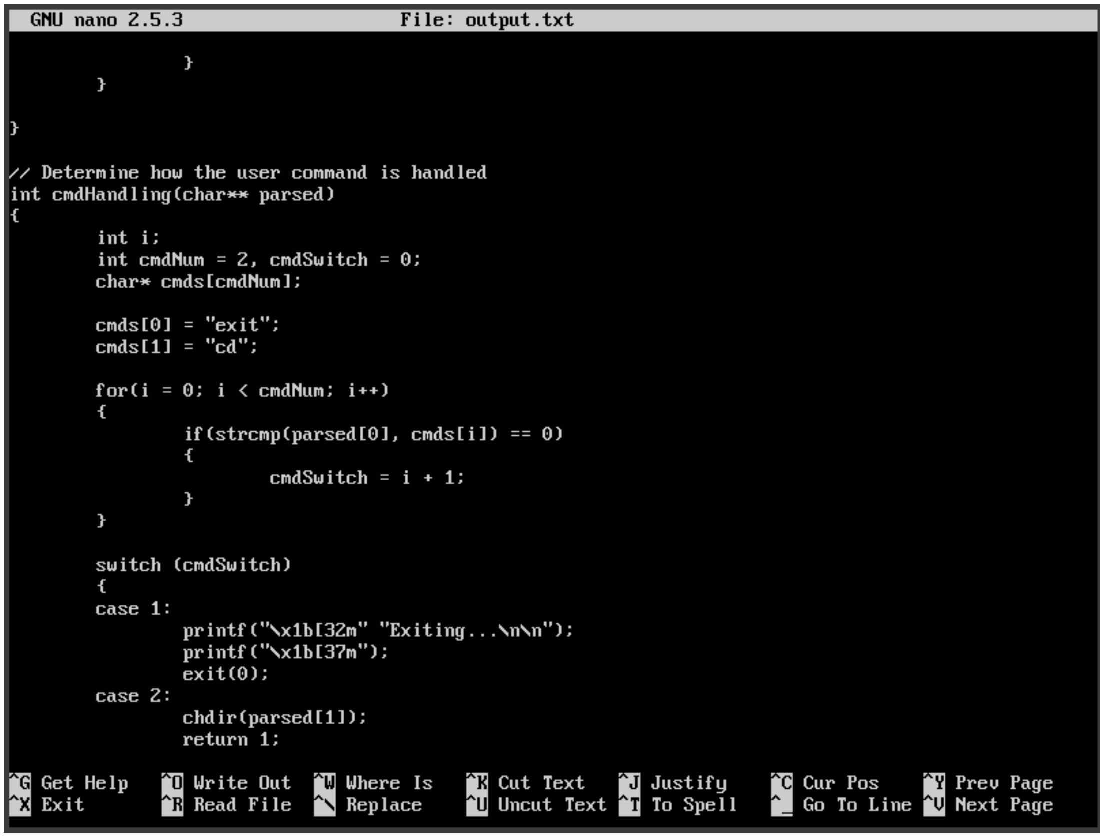
```
  GNU nano 2.5.3                    File: output.txt

-rw-r--r-- 1 aardon aardon 3771 Aug 31  2015 .bashrc
drwx------ 2 aardon aardon 4096 Oct 23 08:40 .cache
drwxrwxr-x 2 aardon aardon 4096 Oct 23 08:49 .nano
-rw-rw-r-- 1 aardon aardon    0 Oct 23 10:18 output.txt
-rw-r--r-- 1 aardon aardon  655 May 16  2017 .profile
^[[37m^[[32m
-----------Program Ended-----------
^[[37m^[[34m/home/aardon> ^[[37mcat /home/osc/dinoBash/dinoBash.c
^[[32m
----------Program Started----------
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <string.h>
#include <curses.h>

// Prints the message of the day
void motd()
{
        printf("\x1b[36m" "\n***** Message of the Day *****\n");
        printf("   Have a wonderful day =)\n");
        printf("******************************\n\n" "\x1b[37m");
}

// Prints the current directory
void currentDir()
{
        char cwd[1024];
        getcwd(cwd, sizeof(cwd));
        printf("\x1b[34m");
        printf("%s> ", cwd);

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  ^V Next Page
```



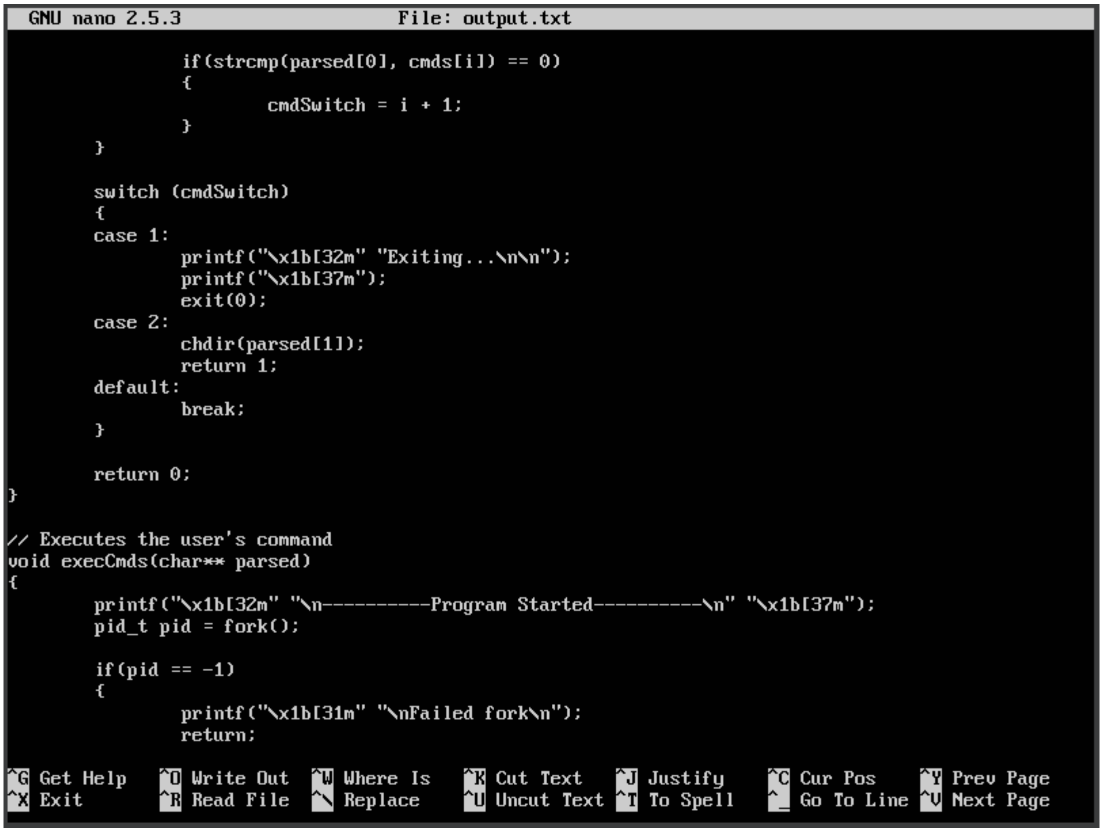
```
// Prints the message of the day
void motd()
{
        printf("\x1b[36m" "\n***** Message of the Day *****\n");
        printf("   Have a wonderful day =)\n");
        printf("******************************\n\n" "\x1b[37m");
}

// Prints the current directory
void currentDir()
{
        char cwd[1024];
        getcwd(cwd, sizeof(cwd));
        printf("\x1b[34m");
        printf("%s> ", cwd);
        printf("\x1b[37m");
}

// Parse the input string from user
void parseString(char* input, char** parsed)
{
        int i;

        for(i = 0; i < 100;i++)
        {
                parsed[i] = strsep(&input, " ");

                if(parsed[i] == NULL)
                {
                        break;
                }
                if(strlen(parsed[i]) == 0)
```



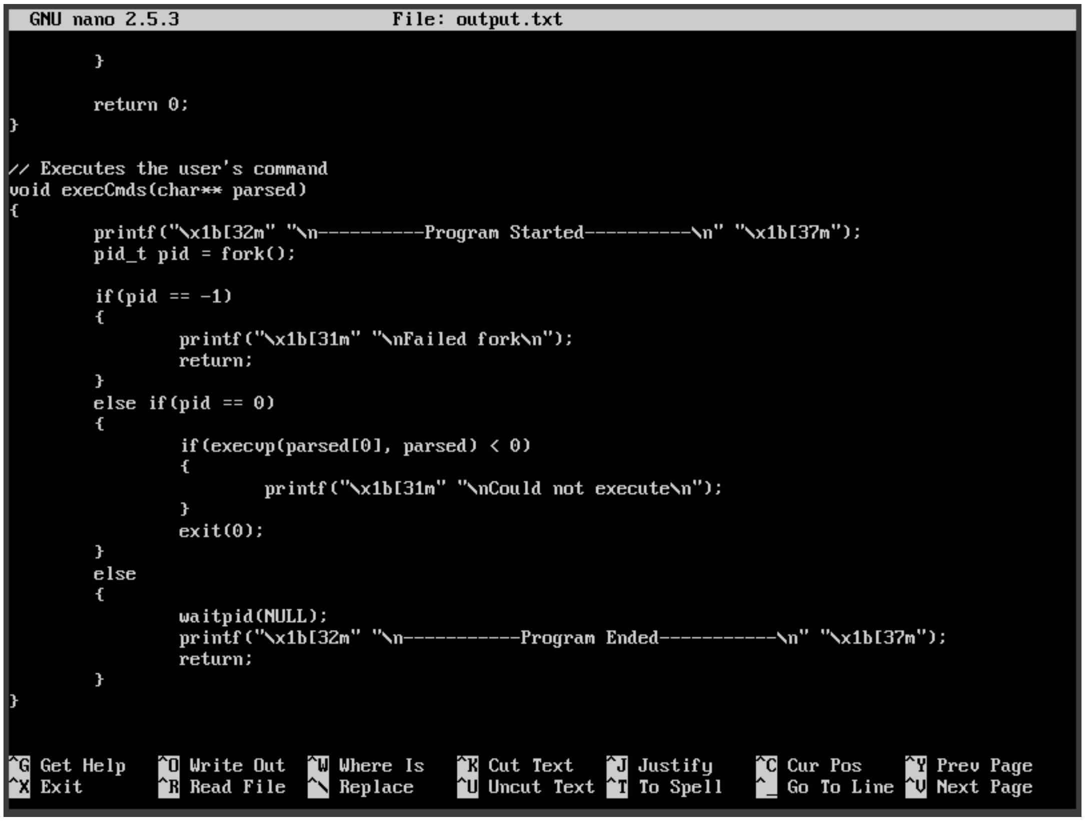
```
// Parse the input string from user
void parseString(char* input, char** parsed)
{
        int i;

        for(i = 0; i < 100;i++)
        {
                parsed[i] = strsep(&input, " ");

                if(parsed[i] == NULL)
                {
                        break;
                }
                if(strlen(parsed[i]) == 0)
                {
                        i--;
                }
        }

}

// Determine how the user command is handled
int cmdHandling(char** parsed)
{
        int i;
        int cmdNum = 2, cmdSwitch = 0;
        char* cmds[cmdNum];

        cmds[0] = "exit";
        cmds[1] = "cd";
```



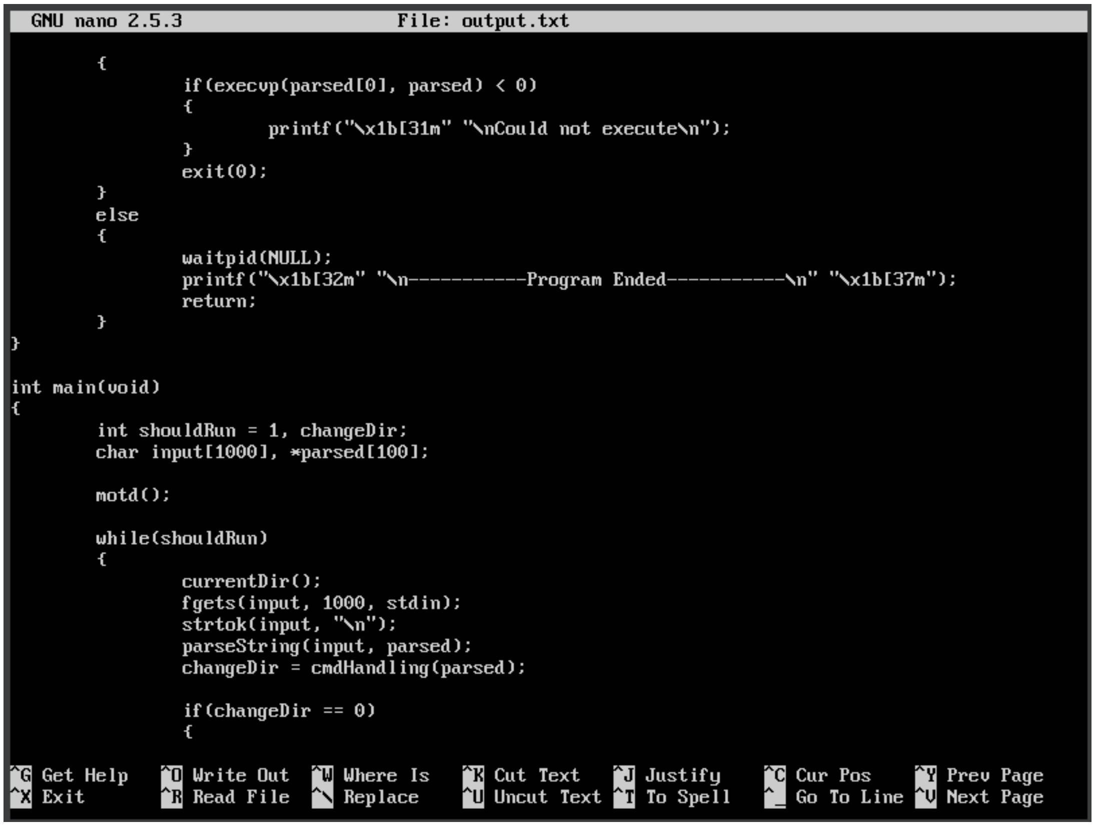
```
                }
        }

}

// Determine how the user command is handled
int cmdHandling(char** parsed)
{
        int i;
        int cmdNum = 2, cmdSwitch = 0;
        char* cmds[cmdNum];

        cmds[0] = "exit";
        cmds[1] = "cd";

        for(i = 0; i < cmdNum; i++)
        {
                if(strcmp(parsed[0], cmds[i]) == 0)
                {
                        cmdSwitch = i + 1;
                }
        }

        switch (cmdSwitch)
        {
        case 1:
                printf("\x1b[32m" "Exiting...\n\n");
                printf("\x1b[37m");
                exit(0);
        case 2:
                chdir(parsed[1]);
                return 1;
```



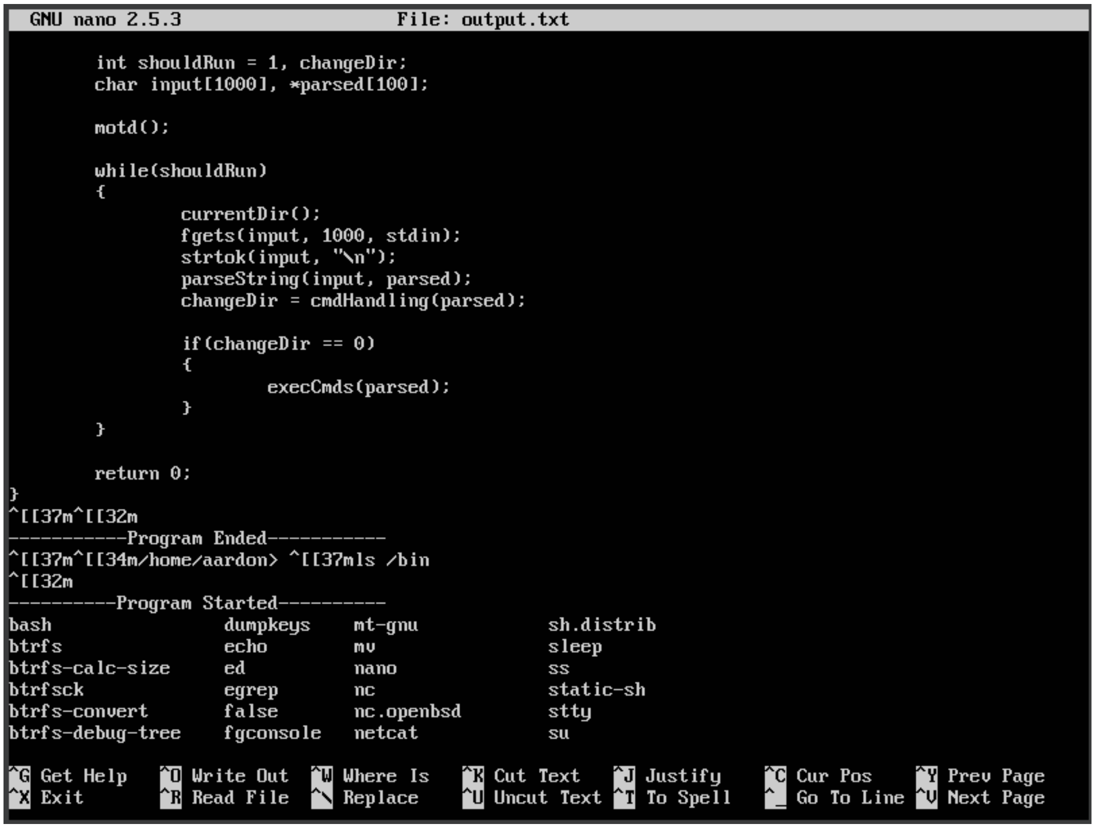
```
                if(strcmp(parsed[0], cmds[i]) == 0)
                {
                        cmdSwitch = i + 1;
                }
        }

        switch (cmdSwitch)
        {
        case 1:
                printf("\x1b[32m" "Exiting...\n\n");
                printf("\x1b[37m");
                exit(0);
        case 2:
                chdir(parsed[1]);
                return 1;
        default:
                break;
        }

        return 0;
}

// Executes the user's command
void execCmds(char** parsed)
{
        printf("\x1b[32m" "\n----------Program Started----------\n" "\x1b[37m");
        pid_t pid = fork();

        if(pid == -1)
        {
                printf("\x1b[31m" "\nFailed fork\n");
                return;
```



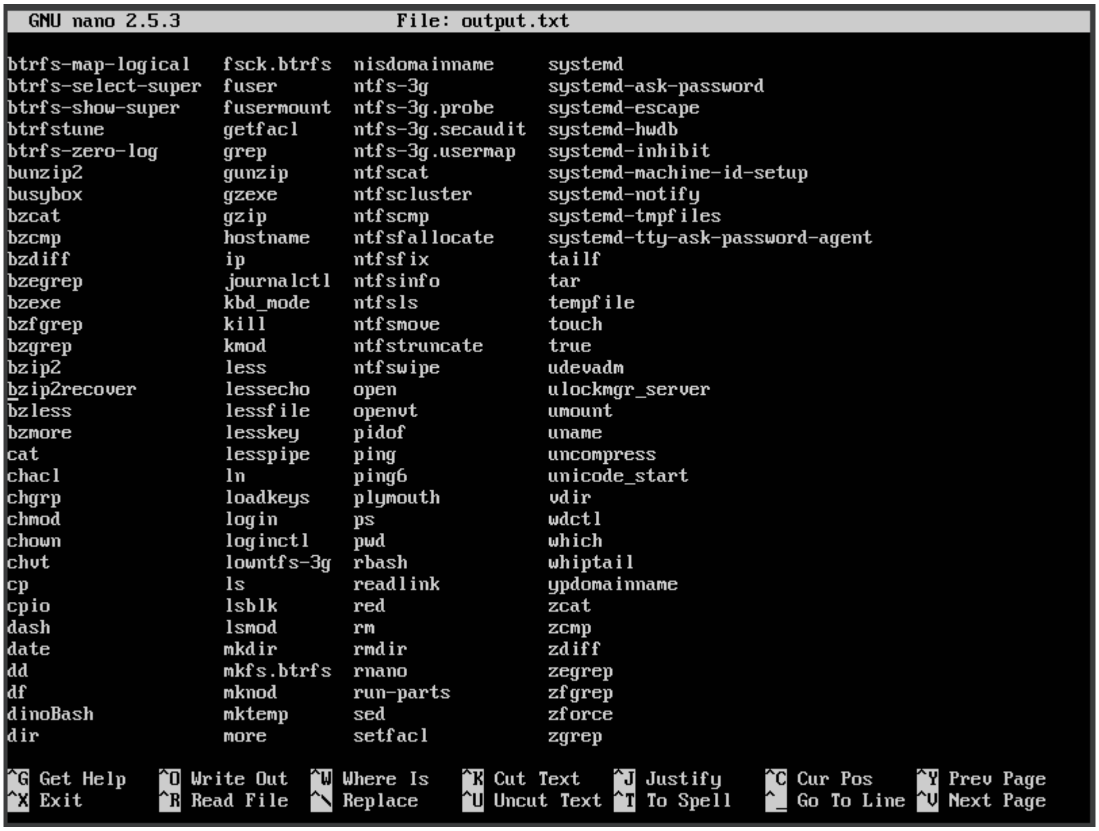
```
        }

        return 0;
}

// Executes the user's command
void execCmds(char** parsed)
{
        printf("\x1b[32m" "\n----------Program Started----------\n" "\x1b[37m");
        pid_t pid = fork();

        if(pid == -1)
        {
                printf("\x1b[31m" "\nFailed fork\n");
                return;
        }
        else if(pid == 0)
        {
                if(execvp(parsed[0], parsed) < 0)
                {
                        printf("\x1b[31m" "\nCould not execute\n");
                }
                exit(0);
        }
        else
        {
                waitpid(NULL);
                printf("\x1b[32m" "\n-----------Program Ended-----------\n" "\x1b[37m");
                return;
        }
}
```



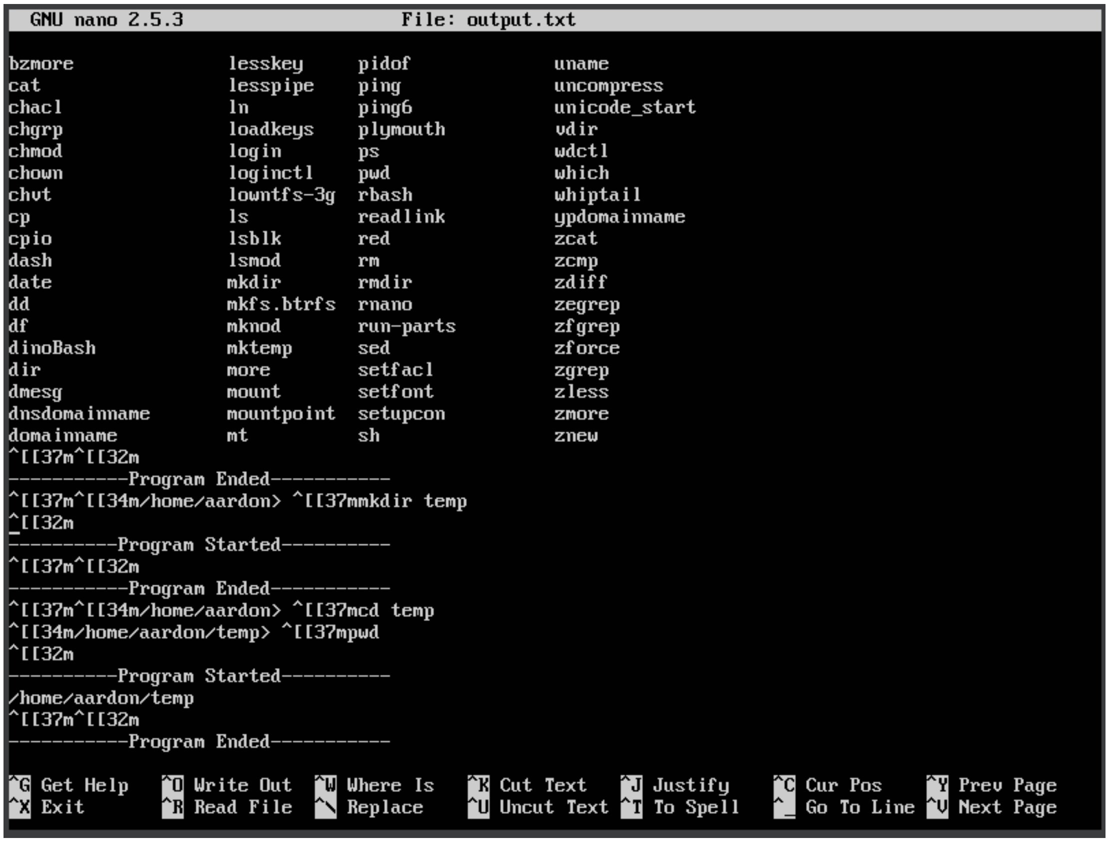
```
        {
                if(execvp(parsed[0], parsed) < 0)
                {
                        printf("\x1b[31m" "\nCould not execute\n");
                }
                exit(0);
        }
        else
        {
                waitpid(NULL);
                printf("\x1b[32m" "\n-----------Program Ended-----------\n" "\x1b[37m");
                return;
        }
}

int main(void)
{
        int shouldRun = 1, changeDir;
        char input[1000], *parsed[100];

        motd();

        while(shouldRun)
        {
                currentDir();
                fgets(input, 1000, stdin);
                strtok(input, "\n");
                parseString(input, parsed);
                changeDir = cmdHandling(parsed);

                if(changeDir == 0)
                {
```



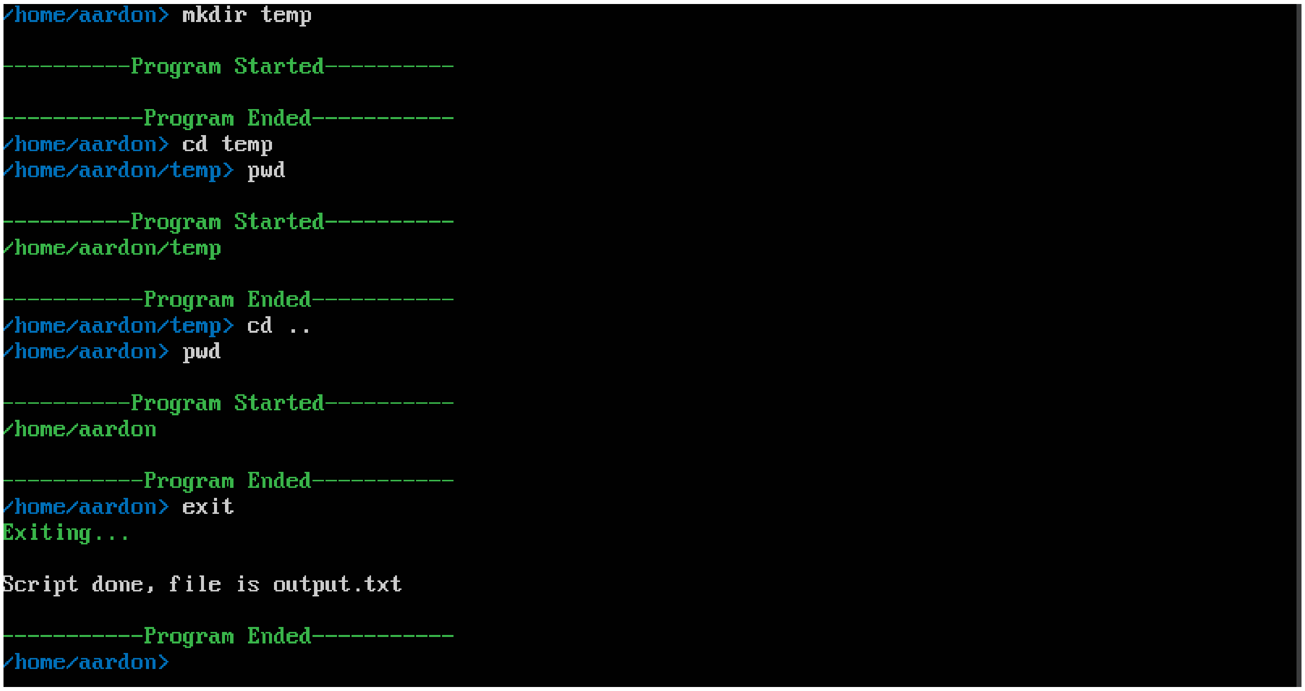
```
        int shouldRun = 1, changeDir;
        char input[1000], *parsed[100];

        motd();

        while(shouldRun)
        {
                currentDir();
                fgets(input, 1000, stdin);
                strtok(input, "\n");
                parseString(input, parsed);
                changeDir = cmdHandling(parsed);

                if(changeDir == 0)
                {
                        execCmds(parsed);
                }
        }

        return 0;
}
^[[37m^[[32m
-----------Program Ended-----------
^[[37m^[[34m/home/aardon> ^[[37mls /bin
^[[32m
----------Program Started----------
bash                dumpkeys    mt-gnu            sh.distrib
btrfs               echo        mv                sleep
btrfs-calc-size     ed          nano              ss
btrfsck             egrep       nc                static-sh
btrfs-convert       false       nc.openbsd        stty
btrfs-debug-tree    fgconsole   netcat            su
```



```
btrfs-map-logical   fsck.btrfs  nisdomainname     systemd
btrfs-select-super  fuser       ntfs-3g           systemd-ask-password
btrfs-show-super    fusermount  ntfs-3g.probe     systemd-escape
btrfstune           getfacl     ntfs-3g.secaudit  systemd-hwdb
btrfs-zero-log      grep        ntfs-3g.usermap   systemd-inhibit
bunzip2             gunzip      ntfscat           systemd-machine-id-setup
busybox             gzexe       ntfscluster       systemd-notify
bzcat               gzip        ntfscmp           systemd-tmpfiles
bzcmp               hostname    ntfsfallocate     systemd-tty-ask-password-agent
bzdiff              ip          ntfsfix           tailf
bzegrep             journalctl  ntfsinfo          tar
bzexe               kbd_mode    ntfsls            tempfile
bzfgrep             kill        ntfsmove          touch
bzgrep              kmod        ntfstruncate      true
bzip2               less        ntfswipe          udevadm
bzip2recover        lessecho    open              ulockmgr_server
bzless              lessfile    openvt            umount
bzmore              lesskey     pidof             uname
cat                 lesspipe    ping              uncompress
chacl               ln          ping6             unicode_start
chgrp               loadkeys    plymouth          vdir
chmod               login       ps                wdctl
chown               loginctl    pwd               which
chvt                lowntfs-3g  rbash             whiptail
cp                  ls          readlink          ypdomainname
cpio                lsblk       red               zcat
dash                lsmod       rm                zcmp
date                mkdir       rmdir             zdiff
dd                  mkfs.btrfs  rnano             zegrep
df                  mknod       run-parts         zfgrep
dinoBash            mktemp      sed               zforce
dir                 more        setfacl           zgrep
```



```
  GNU nano 2.5.3                    File: output.txt

bzmore              lesskey     pidof             uname
cat                 lesspipe    ping              uncompress
chacl               ln          ping6             unicode_start
chgrp               loadkeys    plymouth          vdir
chmod               login       ps                wdctl
chown               loginctl    pwd               which
chvt                lowntfs-3g  rbash             whiptail
cp                  ls          readlink          ypdomainname
cpio                lsblk       red               zcat
dash                lsmod       rm                zcmp
date                mkdir       rmdir             zdiff
dd                  mkfs.btrfs  rnano             zegrep
df                  mknod       run-parts         zfgrep
dinoBash            mktemp      sed               zforce
dir                 more        setfacl           zgrep
dmesg               mount       setfont           zless
dnsdomainname       mountpoint  setupcon          zmore
domainname          mt          sh                znew
^[[37m^[[32m
-----------Program Ended-----------
^[[37m^[[34m/home/aardon> ^[[37mmkdir temp
^[[32m
----------Program Started----------
^[[37m^[[32m
-----------Program Ended-----------
^[[37m^[[34m/home/aardon> ^[[37mcd temp
^[[34m/home/aardon/temp> ^[[37mpwd
^[[32m
----------Program Started----------
/home/aardon/temp
^[[37m^[[32m
-----------Program Ended-----------

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    ^Y Prev Page
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line ^V Next Page
```



```
/home/aardon> mkdir temp

----------Program Started----------

-----------Program Ended-----------
/home/aardon> cd temp
/home/aardon/temp> pwd

----------Program Started----------
/home/aardon/temp

-----------Program Ended-----------
/home/aardon/temp> cd ..
/home/aardon> pwd

----------Program Started----------
/home/aardon

-----------Program Ended-----------
/home/aardon> exit
Exiting...

Script done, file is output.txt

-----------Program Ended-----------
/home/aardon>
```