

Name: tony fayez

ID:5327 gp:2 sec :2

Functions

Remove_endofline

“take the array to remove the final null “

Read_line

“take the line to read it and check if it was exit or not “

Process_line

“taking the line and put every char in its own place in array called temp”

Check_line

“takes the array temp which has the char’s of the command line and Check if there is input or output redirection or if there is & to make a different processes in each case”

Pipe_and_redirection_check

“change the flags and make a processes if there was any of input/output redirection or piping or &”

Read_parse_line

“it is the main function which call all another functions to do there jobs and return to this function to return to int main”

Activities You have the **Auto capture keyboard** option turned on. This will cause the Virtual Machine to automatically **capture** the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the keyboard is captured, you will not be able to type in any other application windows. **main.c [simple shell] - Code::Blocks 20.03**


```
File Edit View Search Project Build Debug Tools Plugins Settings Help
23:20:29.344
main.c [simple shell] - Code::Blocks 20.03

simple shell
ls
ls -l
total 56
-rw-rw-r-- 1 tony tony 95  23:09 b
drwxr-xr-x 3 tony tony 4096 22:17:52 bin
-rw-rw-r-- 1 tony tony 0  24:17:45 h
-rw-rw-r-- 1 tony tony 0  23:09 le
-rw-rw-r-- 1 tony tony 13795 23:20 logfile.log
-rw-rw-r-- 1 tony tony 4850 23:09 main.c
drwxr-xr-x 3 tony tony 4096 22:17:52 obj
-rw-rw-r-- 1 tony tony 1022 22:17:09 'simple shell.cbp'
-rw-rw-r-- 1 tony tony 176 23:21:12 'simple shell.depend'
-rw-rw-r-- 1 tony tony 368 23:21:30 'simple shell.layout'
-rw-rw-r-- 1 tony tony 78 23:21:41 x
-rw-rw-r-- 1 tony tony 76 24:17:45 y

192 //printf("child process terminated\n");
193 fclose(f);
194 }
195
196 int cd(char *path){
197     return chdir(path);
198 }
199
200 int main()
201 {
202     char* args[MAX_WORD];
203     char line[MAX_CHAR];
204     char* piping_args[MAX_WORD];
205     int pipefd[2];
206     pipe(pipefd);
207
208     while(read_parse_line(args,piping_args,line))
209     {
210         pid_t pid = fork();
211
212         if(pid==0)
213         {
214             ///////////////////////////////////
215             if(input_redirection_flag== 1&& input_file!=NULL)
216                 dup2(open(input_file,O_RDWR|O_CREAT,0777),0);
217             ///////////////////////////////////
218             if(output_redirection_flag== 1&& output_file!=NULL)
219                 dup2(open(output_file,O_RDWR|O_CREAT,0777),1);
220             ///////////////////////////////////
221             exec_in_file(){
222                 logfile.log,"a");
223                 //child process terminated\n");
224                 fclose(f);
225             }
226         }
227     }
228 }
```

simple shell

```
firefox %  
ls  
b h logfile.log obj 'simple shell.depend' x  
bin le main.c 'simple shell.cbp' 'simple shell.layout' y  
ls -l  
total 56  
-rwxrwxr-x 1 tony tony 95 29 23:09 b  
-rwxrwxr-x 3 tony tony 4096 22 17:52 bin  
-rwxrwxr-x 1 tony tony 0 24 17:45 h  
-rwxrwxr-x 1 tony tony 0 29 23:09 le  
-rwxrwxr-x 1 tony tony 13921 29 23:21 logfile.log  
-rwxrwxr-x 1 tony tony 4650 29 23:09 main.c  
-rwxrwxr-x 3 tony tony 4096 22 17:52 obj  
-rwxrwxr-x 1 tony tony 1022 22 17:09 'simple shell.cbp'  
-rwxrwxr-x 1 tony tony 176 29 23:12 'simple shell.depend'  
-rwxrwxr-x 1 tony tony 358 29 21:30 'simple shell.layout'  
-rwxrwxr-x 1 tony tony 78 29 21:41 x  
-rwxrwxr-x 1 tony tony 76 24 17:45 y
```

 Search the Web

Top Sites



youtube



web.whatsa...



wep.whatsa...



cboard.cpro...



stackoverflow



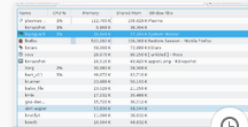
facebook

Highlights



mozilla

More than a browser - Fir...



apps.kde

KSysGuard



stackoverflow

c - Execute commands in ...



Search for anything



```
simple shell
ls
b h logfile.log obj 'simple shell.depend' x
bin le main.c 'simple shell.cbp' 'simple shell.layout' y
ls -l
total 56
-rwxrwxr-x 1 tony tony 95 23:09 b
drwxr-xr-x 3 tony tony 4096 22:17:52 bin
-rwxrwxr-x 1 tony tony 0 23:09 h
-rwxrwxr-x 1 tony tony 0 23:09 le
-rwxrwxr-x 1 tony tony 13796 23:20 logfile.log
-rwxrwxr-x 1 tony tony 4550 23:09 main.c
drwxr-xr-x 3 tony tony 4096 22:17:52 obj
-rwxrwxr-x 1 tony tony 1022 23:09 'simple shell.cbp'
-rwxrwxr-x 1 tony tony 176 23:12 'simple shell.depend'
-rwxrwxr-x 1 tony tony 358 23:13 'simple shell.layout'
-rwxrwxr-x 1 tony tony 78 21:41 x
-rwxrwxr-x 1 tony tony 76 24:17:45 y
firefox
```

Stay ahead of hackers. Check for data breaches with Firefox Monitor.

You've got the web browser that protects your privacy —
now it's time to get a lookout for hackers.

[Check Your Breach Report](#)



Firefox Monitor shows you if your information has been leaked in a known data breach, and alerts you in case it happens in the future.



Stay ahead of hackers