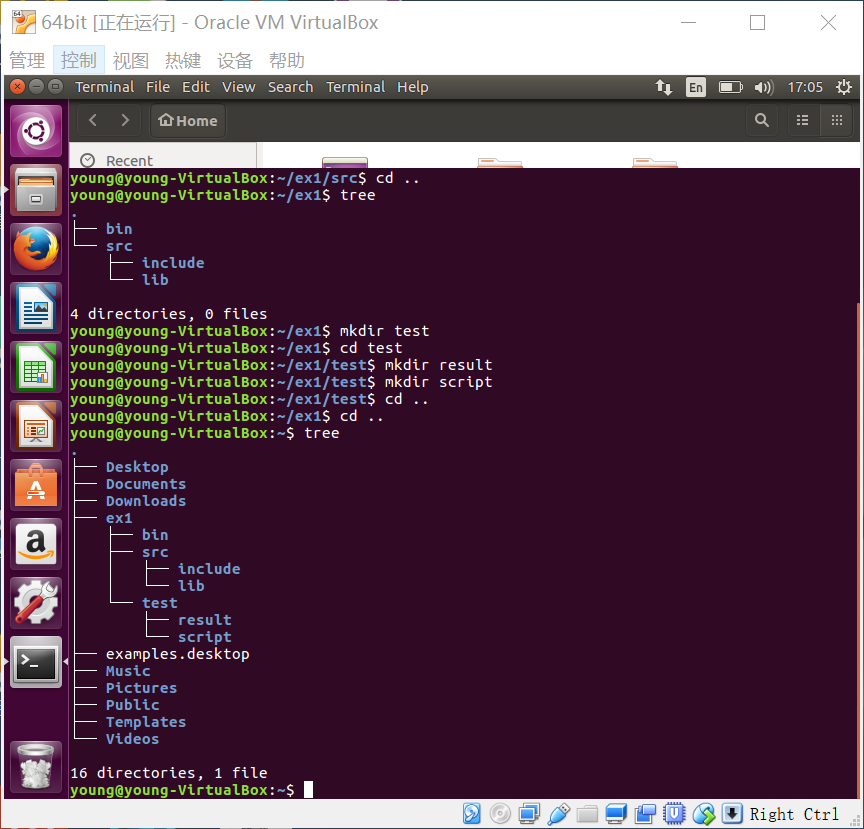
Ex1

Using cd mkdir tree command



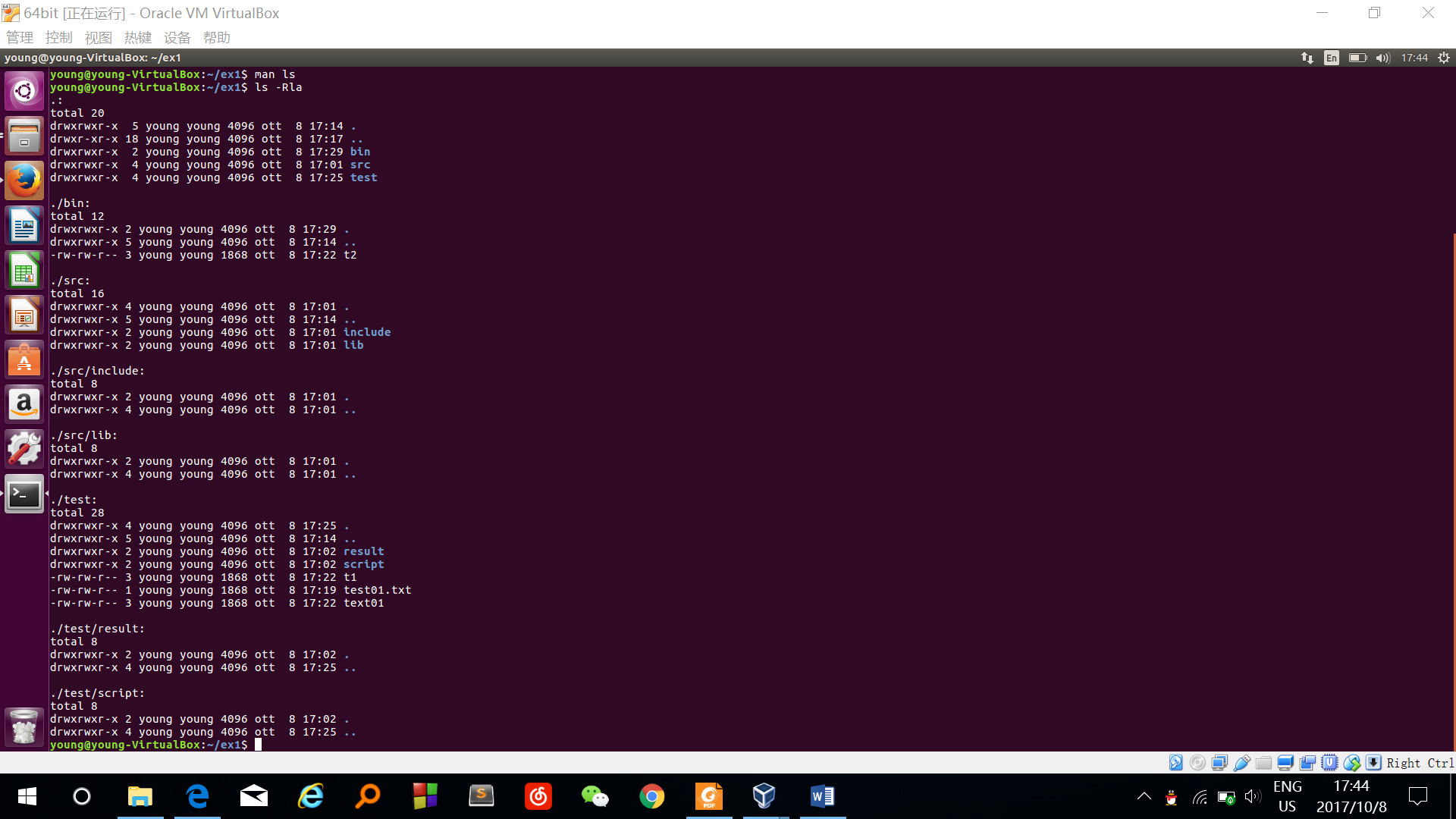
Tab for anto complete

Option -p means no error if existing,make parent dir if needed

e.g.:this ex we can use <mkdir -p ex1/src/include >for efficiency

Ex2

Using cp ln ls -Rla



5. Explain the meaning of the number of links associated to each file or directory.

First block for dir ex1:

. 5 for itself,parent dir,bin,src,test

.. 18 parent dir

bin 2 itself,parent(ex1)

src 4 itself,parent(ex1),include,lib

test 4 itself,parent(ex1),result,script

Second block for ./bin:

. 2 for itself,parent (no link for t2,link in don’t have the link from the parent dir)

.. 5 parent dir(ex1) have 5 link

t2 3 three link to text01(t1,t2,text01)

third block for ./src:

. 2 for itself,parent,include,lib

.. 5 parent dir(ex1) have 5 link

Include 2 itself,parent

Lib 2 itself,parent

And so on…..

That means regular files have link only to them self like text01.txt,t1,t2

Directory files have link to them self and parent and the subdirectory

We can see how many hard link to a file eg.the link to text01.txt is 3

6. Remove file **text01**, and display the content of **t2.** Does it work? Why?

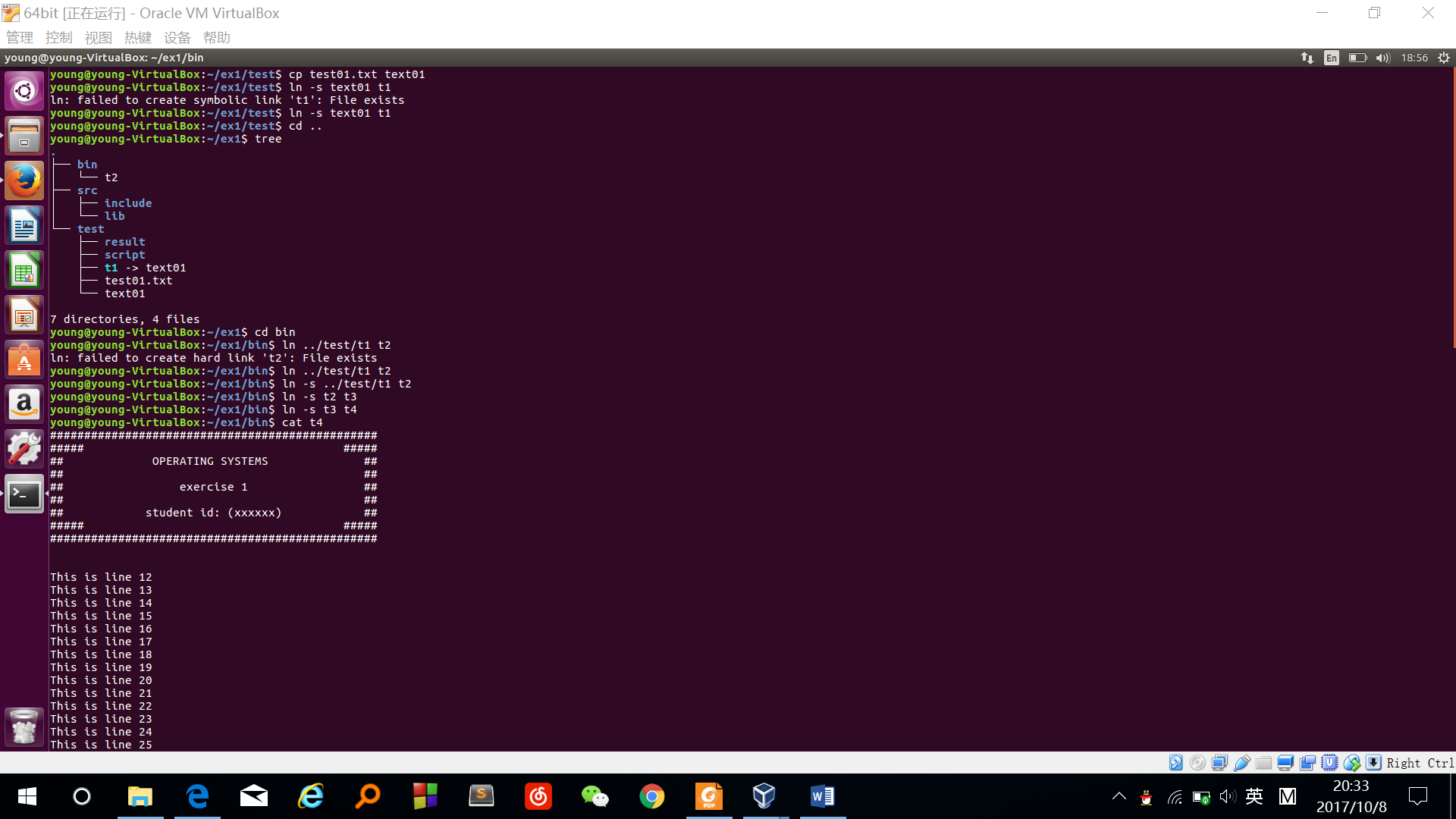
Using rm cat.

it works

Because hard link is like given multiple name to a file. Only delete the file can been delete if all link to the file has been deleted

Ex3

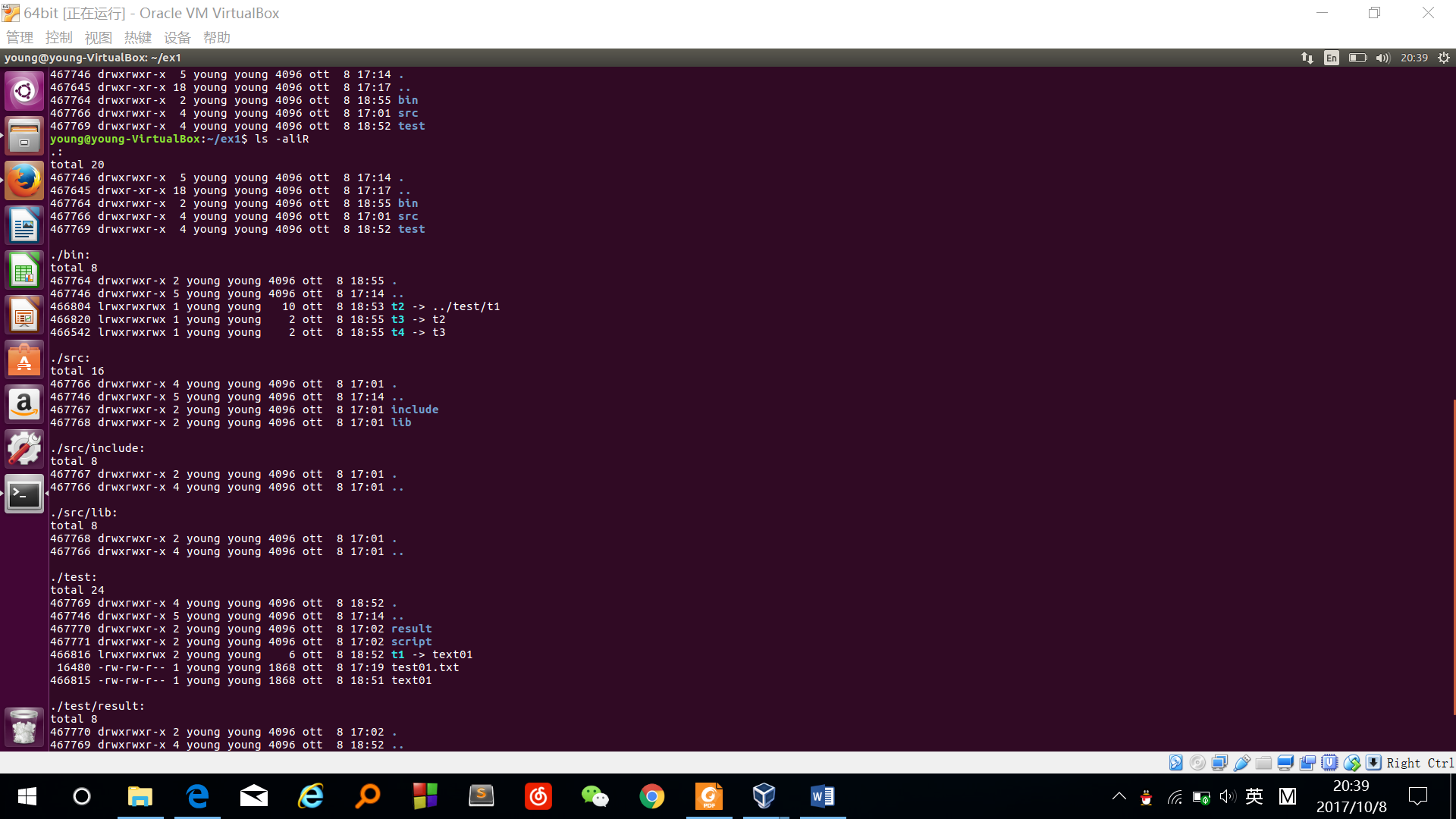
Using ln -s



3. Continue creating soft link **t3** to **t2**, **t4** to **t3**,… , **ti** to **t(i-1)**. Is there any limit?

No limit

4. List all files, and their attributes, in **ex1** and its subdirectories

  
5. Explain the meaning of the number of links associated to each file. ???

T1 2 to test01,txt, t2 to t1

T2,t3,t4 1 itself

6. Remove file **text01**, and display the content of **t4.** Does it work? Why?

Cant display.

No such file

Because t4 is like a pointer to file t3 to t2 to t1 to text01.system don’t know what the refer to is.

7. Remove all file **t** in **bin** with a single command

Rm t\*