**COMP 2766: Introduction to Linux**

**Assignment #7: Regular File and Directory File Permissions**

This assignment consists of two parts. Part 1 has you examine permissions for various directories to see whether you can perform the operations associated with those permissions. Part 2 has you examine the permissions of both regular and directory files and how they interact to control what you can do.

**PART 1: Examining Directory Permissions**

1. Login as the root user or enter: su - root
2. If necessary, enter the following command lines to create a user named after ***your first name and last initial*** (ex: if your name is Justin Trudeau, create a user named justint) and set its password:

# useradd *justint*

# passwd *justint*

1. Enter this command line: su - justint

***You must switch to the user named after your first name and last initial. Failure to do this will result in a mark of 0.***

**\*\*\*\* IMPORTANT: Before continuing, double-check that your command line prompt shows your user name (NOT root) and that your pwd is your user’s home directory (NOT root’s). If you made a mistake entering the above su command line, enter the exit command to close the shell and re-enter the above su command line. \*\*\*\***

1. For several important directories, you will determine the directory’s permissions and what your user can do in that directory. Based on our discussion in class, you will answer some questions and, then, test your answers to see whether they are correct. The first directory has been done for you to show what is expected and let you use it as a model for the other directories that follow.

Let’s start with the root directory. Enter this command line and complete the following table

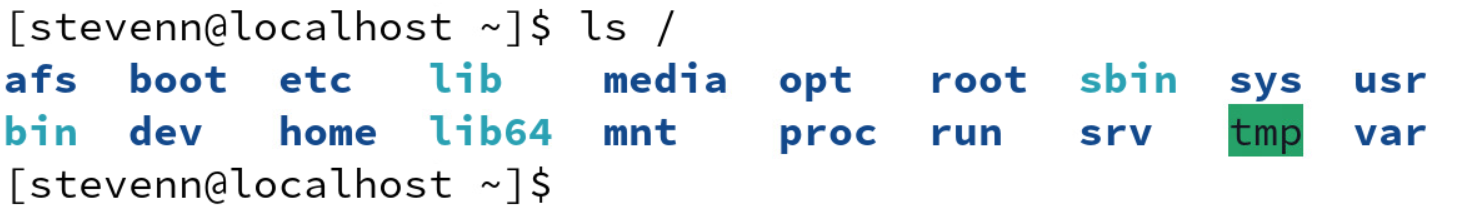
ls -ld /

|  |  |
| --- | --- |
| **Directory file** | / |
| **Permissions for directory file (highlight yours)** | r-xr-xr-x |

**Q:** Can your user do a basic listing of the / directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a basic listing of /** |
| yes | has | r |

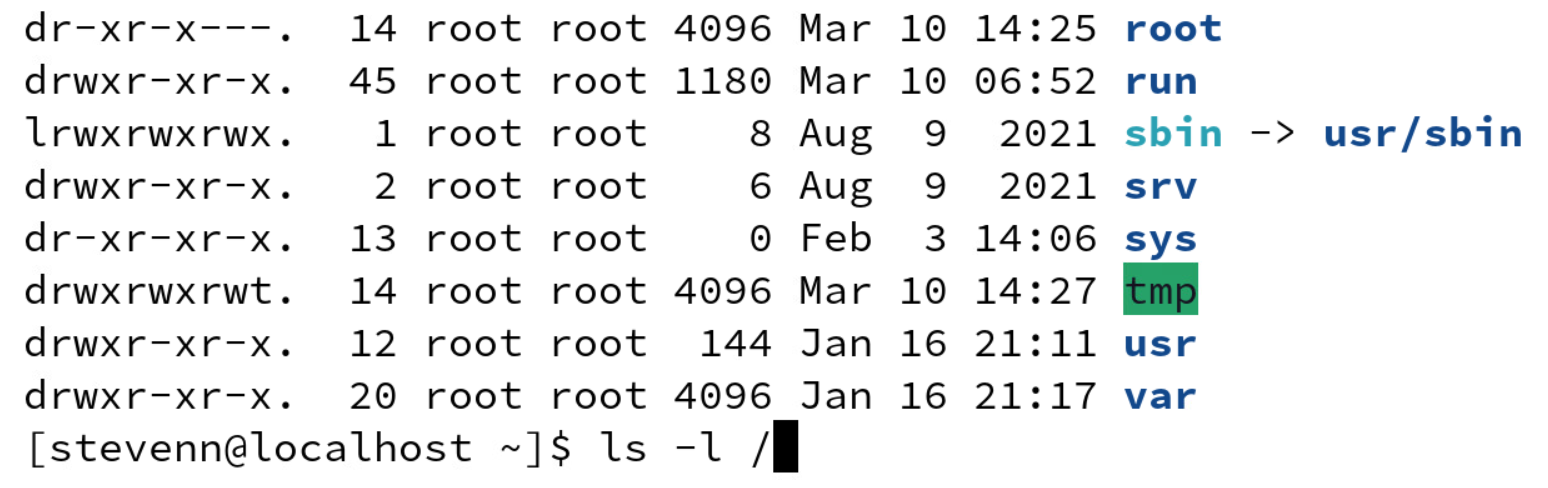
To test your answer, try to do a basic listing of / by entering this command line and insert a screenshot, here, showing the command line and its output: ls /



**Q:** Can your user do a long listing of the / directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a long listing of /** |
| yes | has | r and x |

To test your answer, try to do a long listing of / by entering this command line and insert a screenshot, here, showing the command line and its output: ls -l /



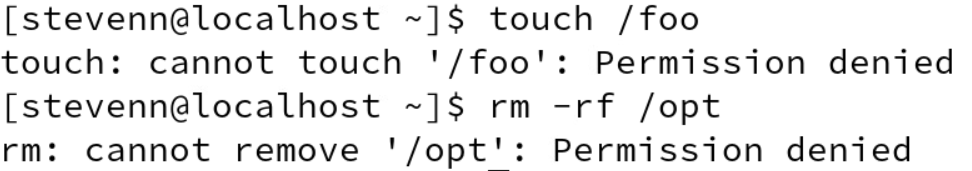
**Q:** Can your user create files in and delete files from the / directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has**  **to create files in and delete files from /** |
| no | lacks | w+x |

To test your answer, try to create a file in and delete a file from / by entering these command lines and insert a screenshot, here, showing these command lines and their output:

touch /foo

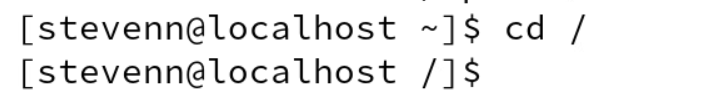
rm -rf /opt



**Q:** Can your user access the / directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to access /** |
| yes | has | x |

To test your answer, try to access / by entering this command line and insert a screenshot, here, showing the command line and its output: cd /



**\*\*\*\* IMPORTANT: When specifying which permission(s) your user has or lacks to do a particular task, specify the MINIMUM permission(s). That is, specify only the permission(s) that your user has or lacks to do the task – NOTHING MORE AND NOTHING LESS. \*\*\*\***

1. (**1 mark**) Using the above example as a model, enter this command line and complete the following table:

ls -ld /home/

|  |  |
| --- | --- |
| **Directory file** | /home/ |
| **Permissions for directory file (highlight yours)** | rwxr-xr-x |

(**1 mark**)**:** Can your user do a basic listing of the /home/ directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a basic listing of /home** |
| Yes | has | r |

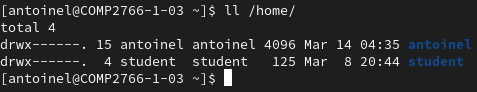
(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: ls /home/



(**1 mark**)**:** Can your user do a long listing of the /home/ directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a long listing of /home** |
| Yes | has | r+x |

(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: ls -l /home/



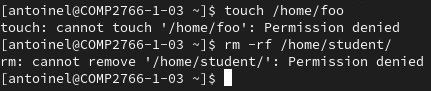
(**1 mark**)**:** Can your user create files in and delete files from the /home/ directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has**  **to create files in and delete files from /home/** |
| No | lacks | w |

(**1 mark**): To test your answer, enter these command lines and insert a screenshot, here, showing these command lines and their output:

touch /home/foo

rm -rf /home/student



(**1 mark**)**:** Can your user access the /home/ directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to access /home/** |
| Yes | has | x |

(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: cd /home/



1. (**1 mark**) Let’s explore your user’s home directory. Enter this command line and complete the following table:

ls -ld ~ ß ~ means the current user’s home directory

|  |  |
| --- | --- |
| **Directory file** | ~ |
| **Permissions for directory file (highlight yours)** | rwx------ |

(**1 mark**)**:** Can your user do a basic listing of the ~ directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a basic listing of ~** |
| Yes | has | r |

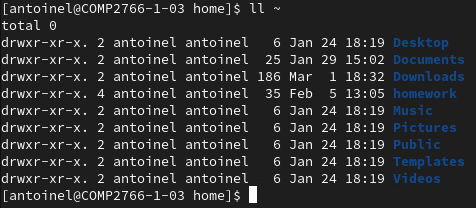
(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: ls ~



(**1 mark**)**:** Can your user do a long listing of the ~ directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a long listing of ~** |
| Yes | has | r+x |

(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: ls -l ~



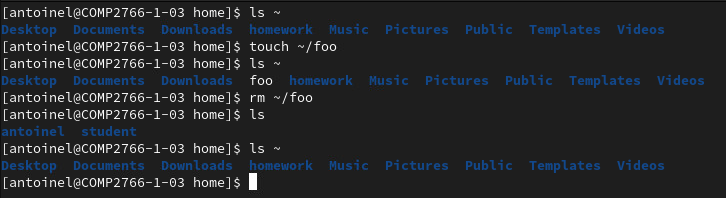
(**1 mark**)**:** Can your user create files in and delete files from the ~ directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has**  **to create files in and delete files from ~** |
| Yes | has | w+x |

(**1 mark**): To test your answer, enter these command lines and insert a screenshot, here, showing these command lines and their output:

touch ~/foo

rm ~/foo



(**1 mark**)**:** Can your user access the ~ directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to access ~** |
| Yes | has | x |

(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: cd ~



1. (**1 mark**) Now, let’s explore the /tmp directory. Enter this command line and complete the following table:

ls -ld /tmp

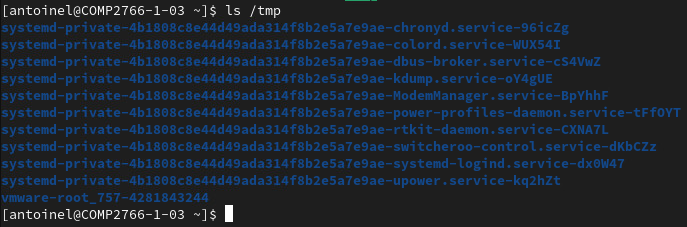
|  |  |
| --- | --- |
| **Directory file** | /tmp |
| **Permissions for directory file (highlight yours)** | rwxrwxrwt |

**NOTE: The t permission is called the sticky bit and will be discussed in an upcoming lecture. If your user has the t permission, you also have the x permission.**

(**1 mark**)**:** Can your user do a basic listing of the /tmp directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a basic listing of /tmp** |
| Yes | has | r |

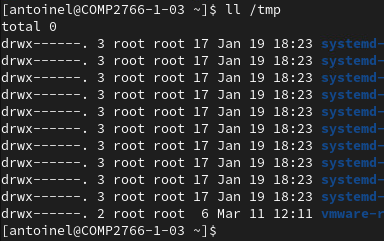
(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: ls /tmp



(**1 mark**)**:** Can your user do a long listing of the /tmp directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a long listing of /tmp** |
| Yes | has | r+x |

(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: ls -l /tmp



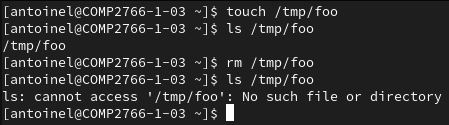
(**1 mark**)**:** Can your user create files in and delete files from the /tmp directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has**  **to create files in and delete files from /tmp** |
| Yes | has | w+x |

(**1 mark**): To test your answer, enter these command lines and insert a screenshot, here, showing these command lines and their output:

touch /tmp/foo

rm /tmp/foo



(**1 mark**)**:** Can your user access the /tmp directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to access /tmp** |
| Yes | has | x |

(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: cd /tmp



1. (**1 mark**) Now, let’s explore /root, the root user’s home directory. Enter this command line and complete the following table:

ls -ld /root

|  |  |
| --- | --- |
| **Directory file** | /root |
| **Permissions for directory file (highlight yours)** | r-xr-x--- |

(**1 mark**)**:** Can your user do a basic listing of the /root directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a basic listing of /root** |
| No | lacks | r |

(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: ls /root



(**1 mark**)**:** Can your user do a long listing of the /root directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to do a long listing of /root** |
| No | lacks | r+x |

(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: ls -l /root



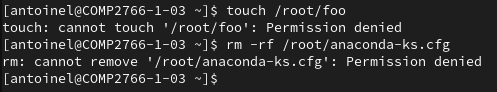
(**1 mark**)**:** Can your user create files in and delete files from the /root directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has**  **to create files in and delete files from /root** |
| No | lacks | w+x |

(**1 mark**): To test your answer, enter these command lines and insert a screenshot, here, showing these command lines and their output:

touch /root/foo

rm /root/anaconda-ks.cfg



(**1 mark**)**:** Can your user access the /root directory? Why (not)?

|  |  |  |
| --- | --- | --- |
| **Yes or No?** | **Lacks or Has?** | **Permission(s) that your user lacks/has to access /root** |
| No | lacks | x |

(**1 mark**): To test your answer, enter this command line and insert a screenshot, here, showing the command line and its output: cd /root



1. **BEFORE PROCEEDING TO PART 2, enter the exit command to close your user’s login shell and revert to the root user**.

**PART 2: Examining the Interaction of Regular and Directory File Permissions**

**PREPARATION FOR PART 2**

1. verify that you are logged in as user root
2. open a new terminal window
3. if necessary, create a user based on your first name (ex: justin) and set the password by entering these command lines:

# useradd *yourFirstName*

# passwd *yourFirstName* (use your first name as the password and ignore any warnings)

1. create a user named elvis and set the password by entering these command lines:

# useradd elvis

# passwd elvis (use elvis as the password and ignore any warnings)

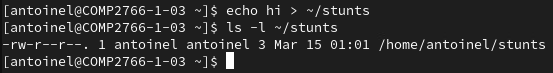
1. switch to your first name user with your user’s login shell by entering: # su – *yourFirstName*
2. in your terminal window, open a new tab (***NOT*** a new terminal window!) by clicking the square with the + symbol in it that is located at the top-left of your terminal window
3. in the new tab, switch to user elvis with a user elvis login shell by entering: # su – elvis
4. as you do this lab, recall that *~username* refers to a user’s home directory (ex: ~elvis means elvis’ home directory)

**VIEWING AND EDITING A REGULAR FILE’S CONTENTS**

1. In your terminal window, click the tab for ***yourFirstName***.
2. As user *yourFirstName*, enter: $ echo hi > ~/stunts
3. Output a long listing of only ~*yourFirstName*/stunts by entering this command line:

$ ls -l ~/stunts

Insert a screenshot showing the command line that you entered and its output, here.



(**1 mark**) Looking ***only*** at the permissions for ~*yourFirstName*/stunts, what can user elvis do with that file? (**NOTE: Ignore any directory permissions**)

ANSWER: read only

1. In your terminal window, click the tab for ***elvis***.
2. As user elvis, enter a command line to try to view the contents of ~*yourFirstName*/stunts. (**HINT**: Do ***NOT*** use the ls command!) ***Insert a screenshot*** showing the command line that you entered and its output, here.



Which permission(s) to which file(s) – regular file and directory file(s) – does user elvis lack that prevents him from viewing the contents of ~*yourFirstName*/stunts?

Fill in the four empty cells in this table using some combination of r, w, x or none:

**NOTE: Be specific when specifying which permission(s) user elvis lacks to view stunts. Specify the minimum permission(s) that he lacks and needs. You may NOT say “lacks all permissions”.**

**\*\*\*\* NOTE: This table has been completed for you to show what is expected and let you use it as a model to complete the tables that follow. The explanations in parentheses are provided to help you understand the answers. Do *NOT* provide explanations in the tables that follow. When specifying which permission(s) your user has or lacks to do a particular task, specify the MINIMUM permission(s). That is, specify only the permission(s) that your user has or lacks to do the task – NOTHING MORE AND NOTHING LESS. \*\*\*\***

|  |  |
| --- | --- |
| **Filename** | **Permission(s) elvis lacks to view stunts (if none, state none)** |
| / | **none** (elvis needs only to access this directory to get to the next one and has the required x permission) |
| /home | **none** (elvis needs only to access this directory to get to the next one and has the required x permission) |
| /home/*yourFirstName* | **x** (elvis cannot access your home directory and, thus, cannot access any files contained within; user elvis does NOT need the r permission to list the contents of your home directory provided that you tell him that there is a file named stunts in it or he can guess that such a file exists) |
| /home/*yourFirstName*/stunts | **none** (elvis has the r permission required to view stunts) |

1. In your terminal window, click the tab for *yourFirstName*.
2. (**1 mark**) As user *yourFirstName*, grant the x permission to ~*yourFirstName* to the other access class (a.k.a. the world), while keeping all other existing permissions as they are. ***Insert a screenshot*** of the command line that you entered and its output, here:



1. In your terminal window, click the tab for ***elvis***.
2. As user elvis, enter a command line to try to view the contents of ~*yourFirstName*/stunts.

***Insert a screenshot*** of the command line that you entered and its output, here.



(**2 marks**) Which permission(s) to which file(s) does user elvis have that enables him to view the contents of ~*yourFirstName*/stunts?

Fill in the four empty cells in this table using some combination of r, w, x:

**NOTE: Be specific when specifying which permission(s) user elvis has to view stunts. Specify the minimum permission(s) that he has and needs. You may NOT say “has all permissions”. Do NOT provide explanations as shown in the table for step #13, above.**

|  |  |
| --- | --- |
| **Filename** | **Permission(s) elvis has that lets him view stunts** |
| / | x |
| /home | x |
| /home/*yourFirstName* | x |
| /home/*yourFirstName*/stunts | r |

1. In user ***elvis***’ tab, enter: echo elvis >> ~*yourFirstName*/stunts

(**2 marks**) Which permission(s) to which file(s) does user elvis lack that prevents him from modifying the contents of ~*yourFirstName*/stunts? Read permission

Fill in the four empty cells in this table using some combination of r, w, x or none:

**NOTE: Be specific when specifying which permission(s) user elvis lacks to edit stunts. Specify the minimum permission(s) that he lacks and needs. You may NOT say “lacks all permissions”.**

|  |  |
| --- | --- |
| **Filename** | **Permission(s) elvis lacks to edit stunts (if none, state none)** |
| / | **none** (elvis needs only to access this directory to get to the next one and has the required x permission) |
| /home | **none** (elvis needs only to access this directory to get to the next one and has the required x permission) |
| /home/*yourFirstName* | **none** (elvis needs only to access this directory to get to the next one and has the required x permission) |
| /home/*yourFirstName*/stunts | Aw (elvis needs the write permission to modify the document and he already has read permission) |

1. In your terminal window, click the tab for *yourFirstName*.
2. (**1 mark**) As user *yourFirstName*, enter a command line that grants the write permission to ~*yourFirstName*/stunts to the other access class (a.k.a. the world), while keeping all other existing permissions as they are. ***Insert a screenshot*** showing the command line that you entered and its output, here:



1. In your terminal window, click the tab for ***elvis***.
2. In user ***elvis***’ tab, enter: echo elvis >> ~*yourFirstName*/stunts

(**2 marks**) Which permission(s) to which file(s) does user elvis have that enables him to modify the contents of ~*yourFirstName*/stunts?

Fill in the four empty cells in this table using some combination of r, w, x:

**NOTE: Be specific when specifying which permission(s) user elvis has to edit stunts. Specify the minimum permission(s) that he has and needs. You may NOT say “has all permissions”.**

|  |  |
| --- | --- |
| **Filename** | **Permission(s) elvis has that lets him edit stunts** |
| / | **x** (elvis only has to access / directory, user elvis does NOT need the r permission to list the contents of your home directory provided that you tell him that there is a file named stunts in it or he can guess that such a file exists) |
| /home | **x** (elvis only has to access /home/ directory, user elvis does NOT need the r permission to list the contents of your home directory provided that you tell him that there is a file named stunts in it or he can guess that such a file exists) |
| /home/*yourFirstName* | **x** (elvis only has to access /home/antoinel directory, user elvis does NOT need the r permission to list the contents of your home directory provided that you tell him that there is a file named stunts in it or he can guess that such a file exists) |
| /home/*yourFirstName*/stunts | w (elvis only needs to be able to modify the document; he doesn’t need to be able to read the file nor execute it) |

**LISTING A DIRECTORY’S CONTENTS**

1. In user ***elvis***’ tab, enter a command line to do a basic listing (ls, ***NOT*** ls –l) of the contents of ~*yourFirstName*. ***Insert a screenshot*** showing the command line that you entered and its output, here:



(**2 marks**) Which permission(s) to which file(s) does elvis lack that prevents him from doing a basic listing of your home directory?

Fill in the three empty cells in this table using some combination of r, w, x or none:

**NOTE: Be specific when specifying which permission(s) user elvis lacks to list your home directory. Specify the minimum permission(s) that he lacks and needs. You may NOT say “lacks all permissions”.**

|  |  |
| --- | --- |
| **Filename** | **Permission(s) elvis lacks to list (if none, state none)** |
| / | **none** (elvis only has to access / directory, user elvis does NOT need the r permission to list the contents of your home directory provided that you tell him that there is a file named stunts in it or he can guess that such a file exists) |
| /home | none (elvis only has to access /home directory, user elvis does NOT need the r permission to list the contents of your home directory provided that you tell him that there is a file named stunts in it or he can guess that such a file exists) |
| /home/*yourFirstName* | r (elvis only has to be able to see what’s in the directory, he doesn’t need to be able to modify the content of its nor execute it) |

1. In your terminal window, click the tab for *yourFirstName*.
2. (**1 mark**) As user *yourFirstName*, enter a command line that grants only the read permission to ~*yourFirstName* to the other access class, while keeping all other existing permissions as they are. ***Insert a screenshot*** showing the command line that you entered and its output, here:



1. In your terminal window, click the tab for ***elvis***.
2. In user ***elvis***’ tab, enter a command line to do a basic listing (ls, *not* ls –l) of the contents of ~*yourFirstName*. Insert a screenshot showing the command line that you entered and its output, here:



(**2 marks**) Which permission(s) to which directory files does elvis have that lets him do a basic listing of your home directory?

Fill in the three empty cells in this table using some combination of r, w, x:

**NOTE: Be specific when specifying which permission(s) user elvis has to list your home directory. Specify the minimum permission(s) that he has and needs. You may NOT say “has all permissions”.**

|  |  |
| --- | --- |
| **Filename** | **Permission(s) elvis has to list (if none, state none)** |
| / | **x** (elvis only has to access / directory, user elvis does NOT need the r permission to list the contents of your home directory provided that you tell him that there is a file named stunts in it or he can guess that such a file exists) |
| /home | **x** (elvis only has to access /home directory, user elvis does NOT need the r permission to list the contents of your home directory provided that you tell him that there is a file named stunts in it or he can guess that such a file exists) |
| /home/*yourFirstName* | r (elvis only has to be able to see what’s in the directory, he doesn’t need to be able to modify the content of its nor execute it) |

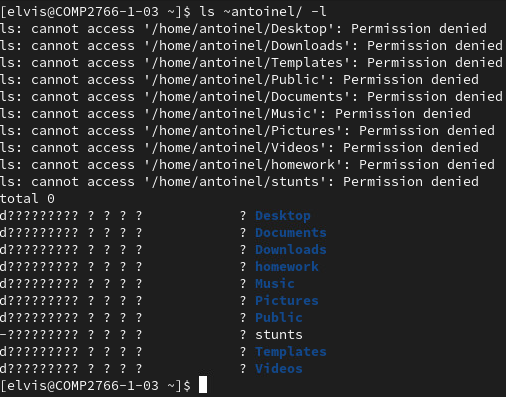
1. (**1 mark**) As user elvis, try to do a long listing (ls –l) of the contents of ~*yourFirstName*. Can he?

ANSWER (yes/no): yes

1. (**1 mark**) As user *yourFirstName*, remove **only** the x permission to ~*yourFirstName* from the other access class (a.k.a. permission group), without changing any other existing permissions. ***Insert a screenshot*** showing the command line that you entered and its output, here:



1. (**1 mark**) As user elvis, try to do a basic listing (ls, *not* ls –l) of the contents of ~*yourFirstName*. It sort of works as you can see the stunts file name in the output, but there is also an error message. It’s sort of like looking into a room through a window, but having no access to the room. ***Insert a screenshot*** showing the command line that you entered and its output, here:



Correction : ls ~antoinel

1. (**1 mark**) As user elvis, try to do a long listing (ls –l) of the contents of ~*yourFirstName*. It doesn’t work, as evidenced by the question marks. To do a long listing of a directory’s contents, one needs the directory’s read and execute permissions. ***Insert a screenshot*** showing the command line that you entered and its output, here:

Correction : ls –l ~antoinel



**REMOVING A DIRECTORY’S CONTENTS**

1. Confirm that the other access class (a.k.a. the world) still has the w permission to ~*yourFirstName*/*stunts*. No screenshot required.
2. (**1 mark**) As user *yourFirstName*, grant the x permission to ~*yourFirstName* to the other access class, without changing any other existing permissions. ***Insert a screenshot*** showing the command line that you entered and its output, here:



1. (**1 mark**) As user elvis, try to remove ~*yourFirstName*/*stunts*. This should fail, as the other access class has no w permission to ~*yourFirstName*. **Though user elvis has the w permission to** *~yourFirstName/stunts***, the w permission for a regular file like *stunts* only controls editing of the file, not its removal**. ***Insert a screenshot*** showing the command line that you entered and its output, here:



1. (**1 mark**) As user *yourFirstName*, remove the w permission for ~*yourFirstName/stunts* from the other access class, without changing any other existing permissions. ***Insert a screenshot*** showing the command line that you entered and its output, here:

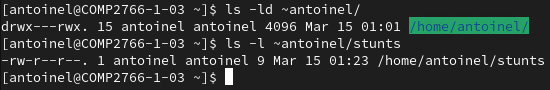


1. (**1 mark**) As user *yourFirstName*, grant the w permission to ~*yourFirstName* to the other access class, without changing any other existing permissions. ***Insert a screenshot*** showing the command line that you entered and its output, here:



1. **As user *yourFirstName***, capture and ***insert ONE screenshot***, here, that contains the output of the following two command lines:

* (**1 mark**) ls –ld ~*yourFirstName*
* (**1 mark**) ls –l ~*yourFirstName/stunts*



1. (**1 mark**) As user elvis, remove ~*yourFirstName/stunts*. This should now work, as others have the w and x permissions to ~*yourFirstName*. ***Insert a screenshot*** showing the command line that you entered and its output, here:

