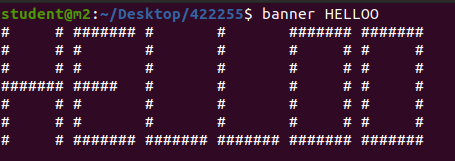
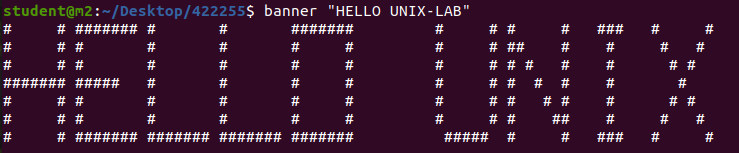
**1. Implement the commands banner, mkdir, rmdir, mv, wc, pwd, ls, rm, man, chmod, chown, chgrp commands along with options for each command.**

**1. BANNER**

* The banner command is used to display characters in a large banner format. Here are examples of using the banner command with different text:
  + **Basic Usage:**
    - Command: banner Hello
    - Example: Display the word "Hello" in a large banner format.
* banner Hello



* **Multiple Words:**
  + Command: banner "Hello World"
  + Example: Display the phrase "Hello World" in a large banner format. Enclosing the text in double quotes is necessary when there are multiple words.
* banner "Hello World"



* **Numbers:**
  + Command: banner 12345
  + Example: Display the numbers "12345" in a large banner format.
* banner 12345



* **Special Characters:**
  + Command: banner "@#$%"
  + Example: Display the special characters "@#$%" in a large banner format.
* banner "@#$%"



* **Custom Character Width:**
  + Command: banner -w 40 Hello
  + Example: Display the word "Hello" in a large banner format with a custom width of 40 characters.
* banner -w 40 Hello



* **Center Text:**
  + Command: banner -w 40 -c Hello
  + Example: Display the word "Hello" in a large banner format with a custom width of 40 characters, centered.  
    bash
* banner -w 40 -c Hello

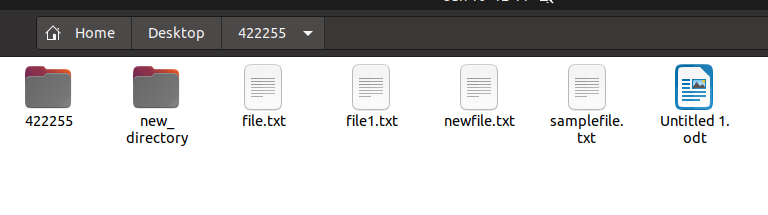


* **Adjusting Height:**
  + Command: banner -w 30 -h 5 Hello
  + Example: Display the word "Hello" in a large banner format with a custom width of 30 characters and a height of 5 lines.
* banner -w 30 -h 5 Hello



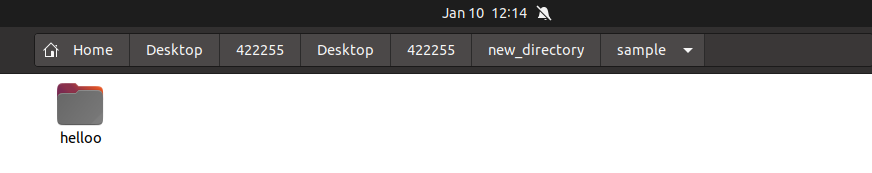
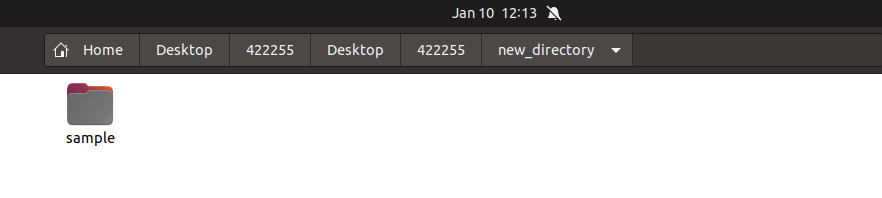
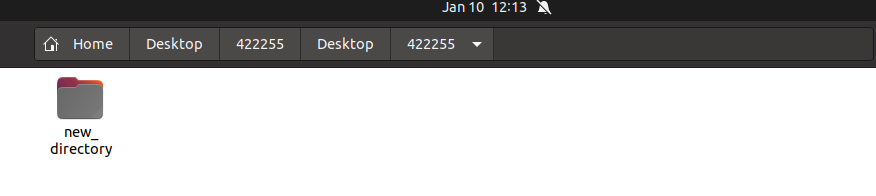
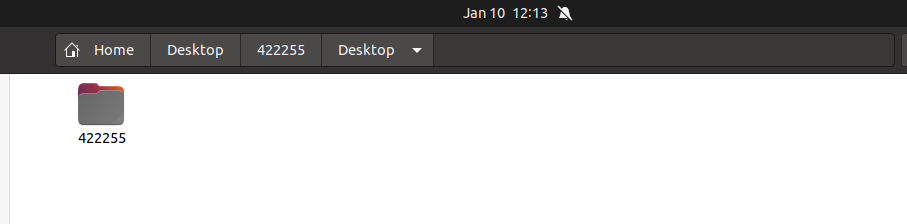
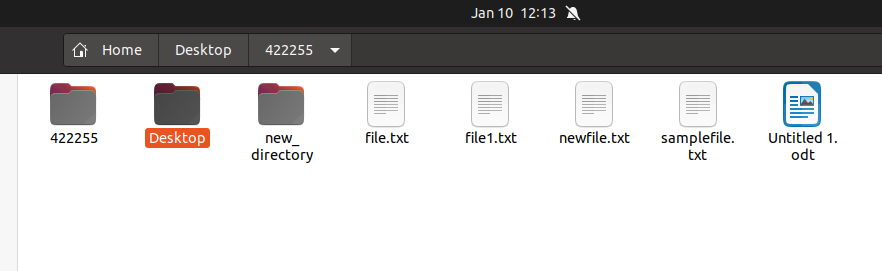
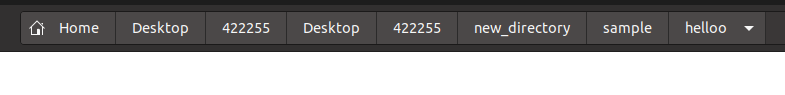
**2. MKDIR**

* The mkdir command is used to create directories. Here are some common options for the mkdir command along with examples:
  + **Basic Usage:**
    - Command: mkdir directory\_name
    - Example: Create a directory named "new\_directory."
* mkdir new\_directory

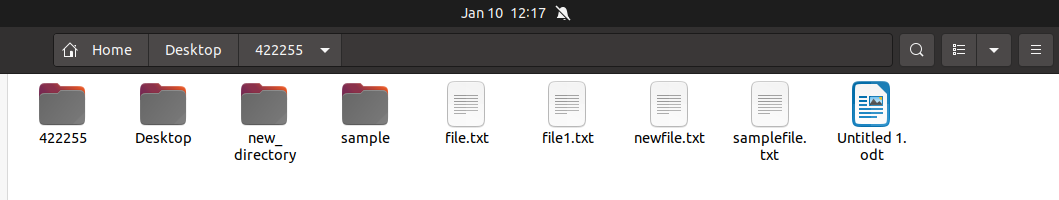


* **Create Parent Directories:**
  + Command: mkdir -p path/to/new/directory
  + Example: Create a directory and its parent directories if they don't exist.  
    bash
* mkdir -p path/to/new/directory

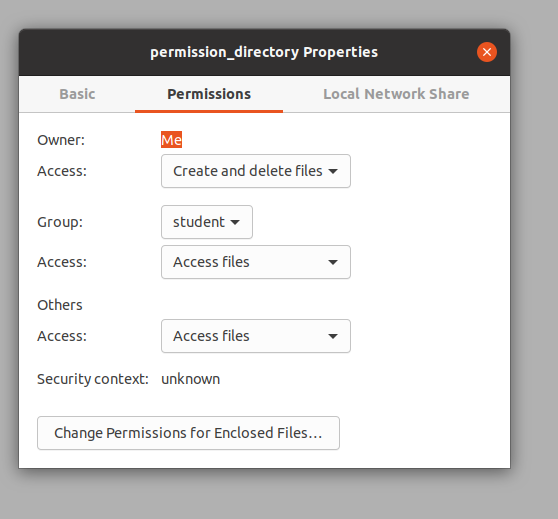
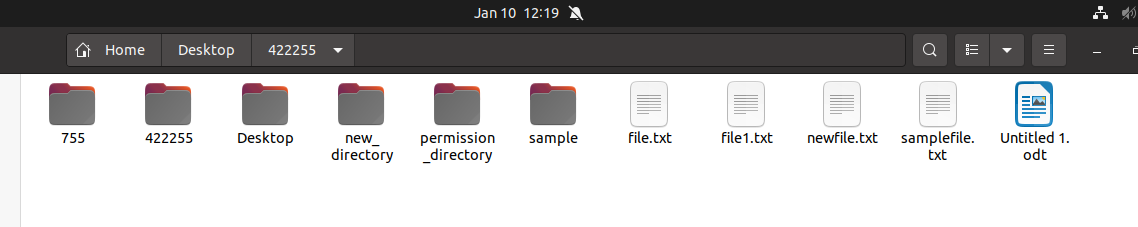




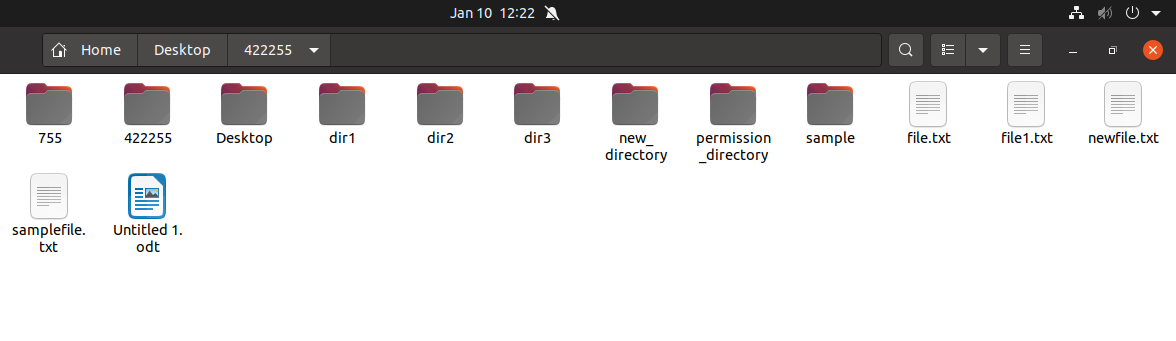
* **Verbose Mode:**
  + Command: mkdir -v directory\_name
  + Example: Display a message for each directory created.
* mkdir -v new\_directory



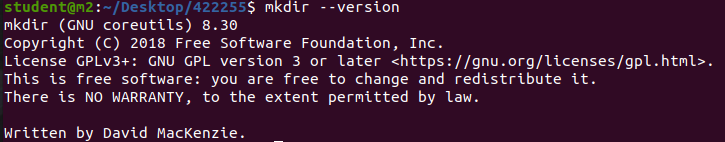
* **Set Permissions:**
  + Command: mkdir -m mode directory\_name
  + Example: Create a directory with specific permissions (e.g., 755).
* mkdir -m 755 new\_directory

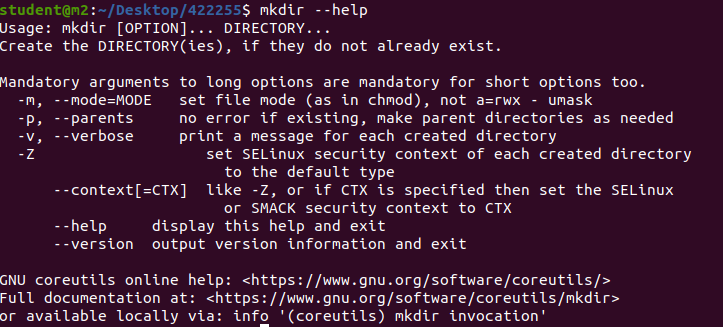


* **Create Multiple Directories:**
  + Command: mkdir dir1 dir2 dir3
  + Example: Create multiple directories in a single command.  
    bash
* mkdir dir1 dir2 dir3



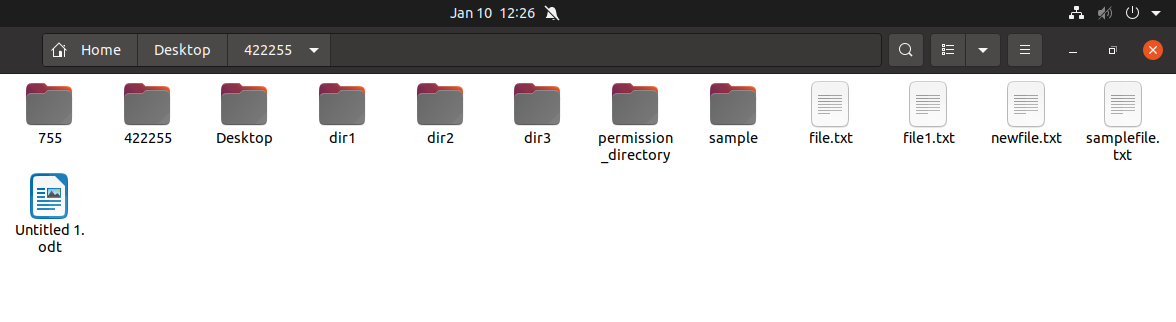
* **Version Information:**
  + Command: mkdir --version
  + Example: Display version information for mkdir.
* mkdir --version



* **Help Information:**
  + Command: mkdir --help
  + Example: Display help information for mkdir.
* mkdir --help

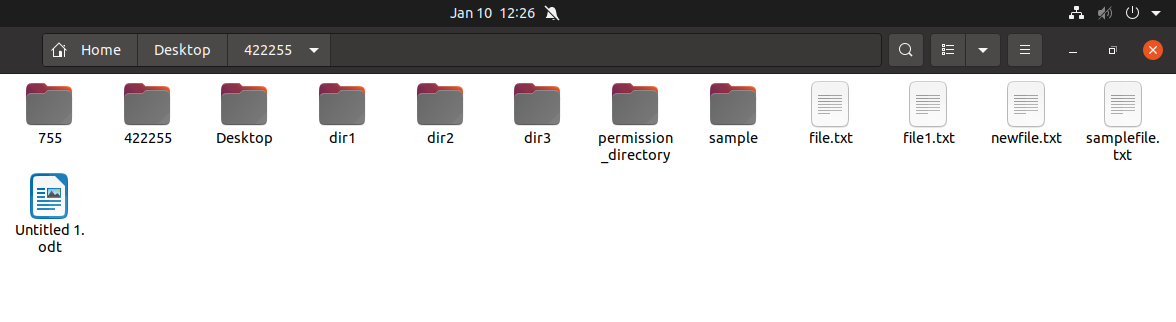
**3. RMDIR**

* The rmdir command is used to remove empty directories. Here are some common options for the rmdir command along with examples:
  + **Basic Usage:**
    - Command: rmdir directory\_name
    - Example: Remove an empty directory named "empty\_directory."
* rmdir empty\_directory

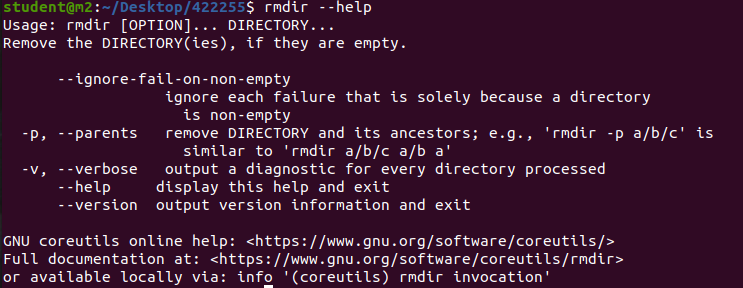


* **Verbose Mode:**
  + Command: rmdir -v directory\_name
  + Example: Display a message for each directory removed.
* rmdir -v empty\_directory



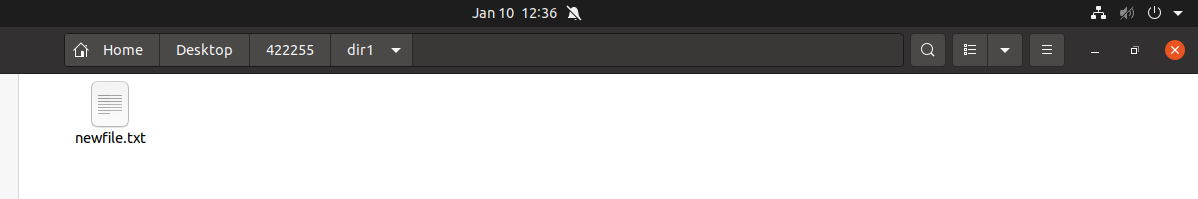


* **Help Information:**
  + Command: rmdir --help
  + Example: Display help information for rmdir.  
    bash
* rmdir --help

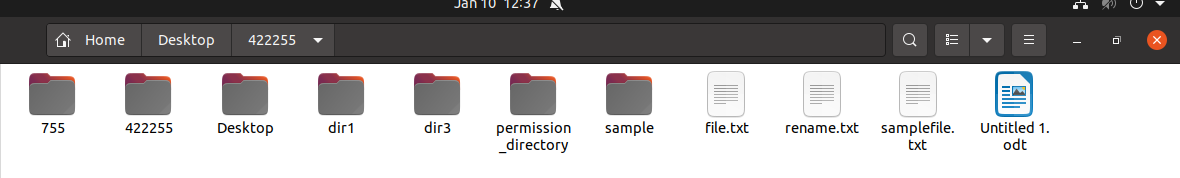
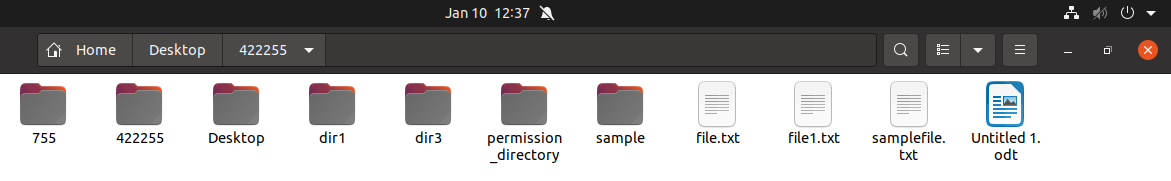


**4. MOVE (mv)**

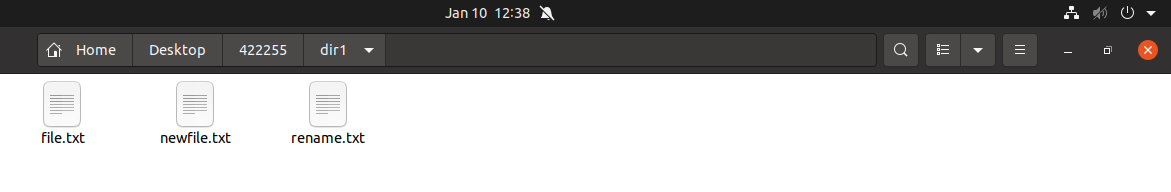
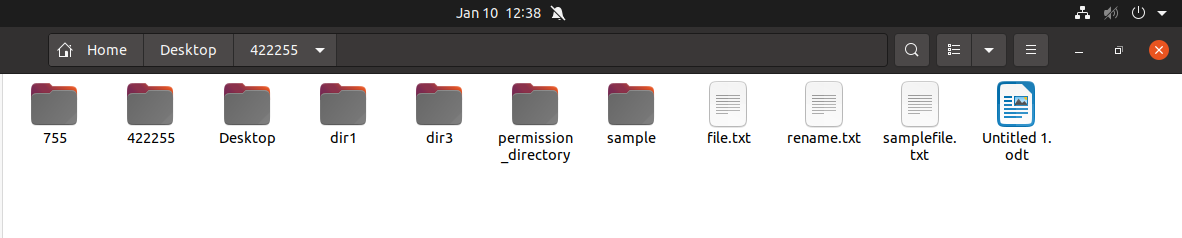
* The mv command is used to move or rename files and directories. Here are some common options for the mv command along with examples:
  + **Basic Usage - Move File:**
    - Command: mv source\_file destination\_directory
    - Example: Move a file named "file.txt" to a directory named "destination."
* mv file.txt destination/



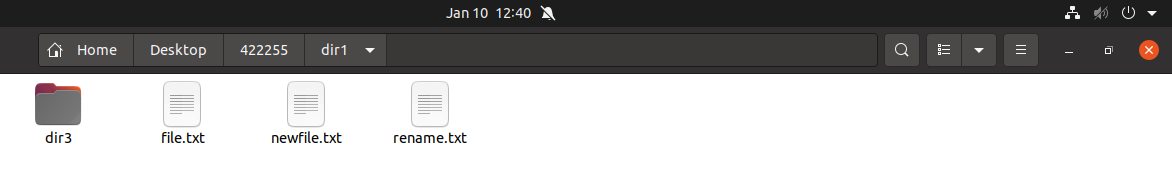
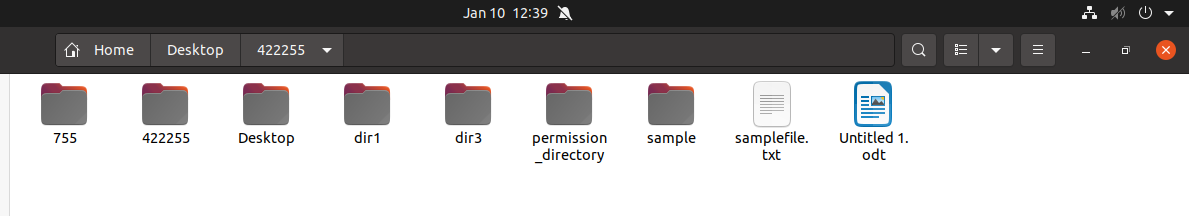
* **Basic Usage - Rename File:**
  + Command: mv old\_filename new\_filename
  + Example: Rename a file from "old.txt" to "new.txt."
* mv old.txt new.txt



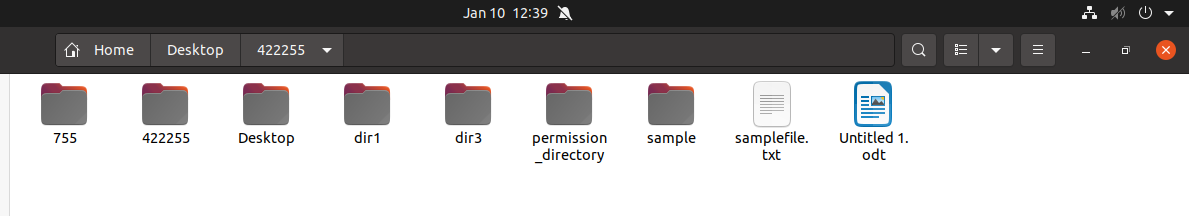
* **Move Multiple Files:**
  + Command: mv file1 file2 directory
  + Example: Move multiple files to a directory.  
    bash
* mv file1 file2 destination/



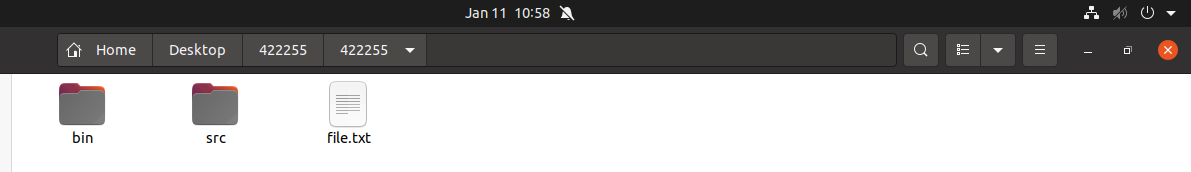
* **Move and Rename Directory:**
  + Command: mv old\_directory new\_directory
  + Example: Rename a directory from "old\_dir" to "new\_dir."
* mv old\_dir new\_dir



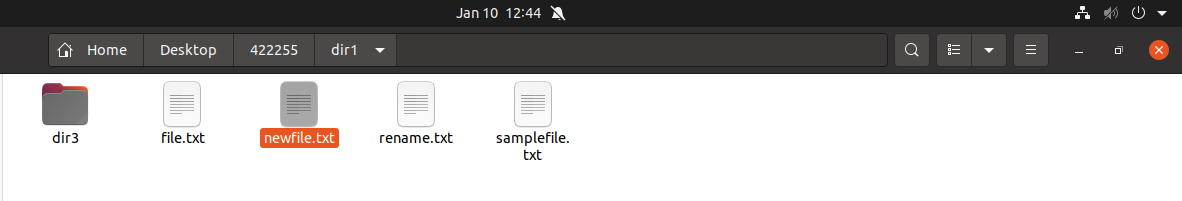
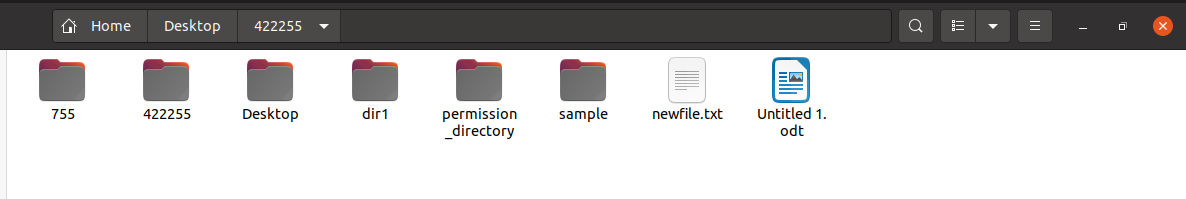
* **Update -u:**
  + Command: mv -u source\_file destination\_directory
  + Example: Move the source file to the destination only if the source is newer or the destination does not exist.  
    bash
* mv -u file.txt destination/



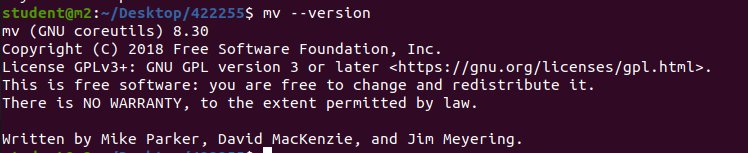
* **Force Move -f:**
  + Command: mv -f source\_file destination\_directory
  + Example: Force move by overwriting existing files without prompting.  
    bash
* mv -f file.txt destination/



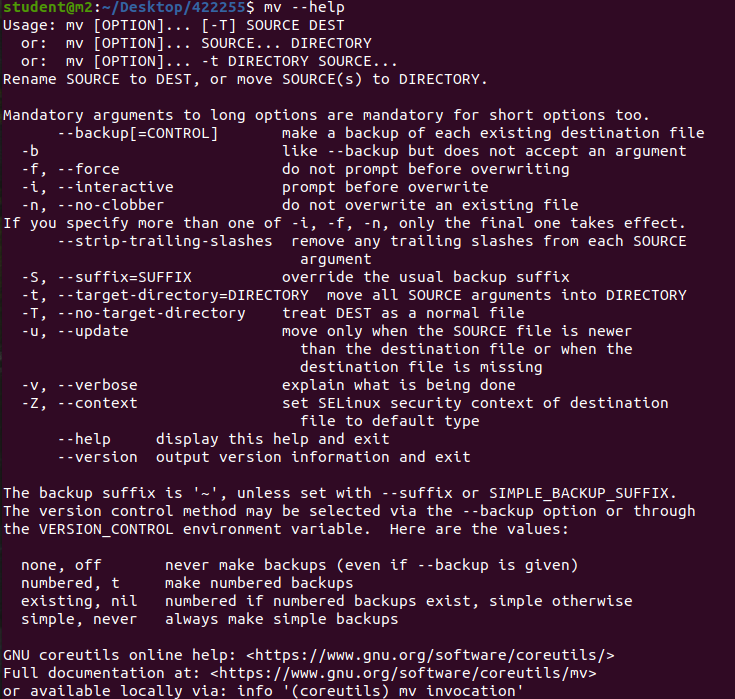
* **Verbose Mode:**
  + Command: mv -v source\_file destination\_directory
  + Example: Display detailed information about the move operation.  
    bash
* mv -v file.txt destination/



* **Version Information:**
  + Command: mv --version
  + Example: Display version information for mv.  
    bash
* mv --version

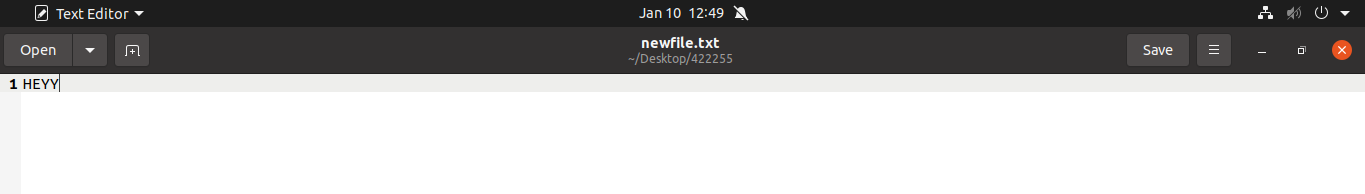


* **Help Information:**
  + Command: mv --help
  + Example: Display help information for mv.
* mv --help



**5. WC**

* The wc command is used to count lines, words, and characters in a file. Here are some common options for the wc command along with examples:
  + **Basic Usage - Count Lines, Words, and Characters:**
    - Command: wc filename
    - Example: Count lines, words, and characters in a file named "example.txt."
* wc example.txt

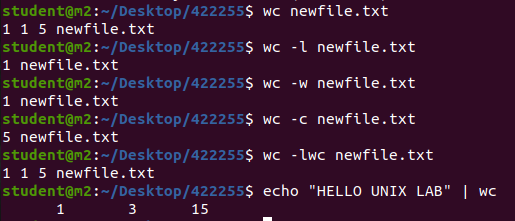
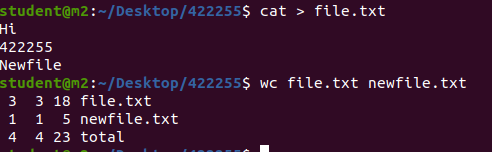


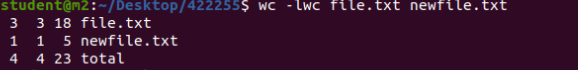
* **Count Lines Only:**
  + Command: wc -l filename
  + Example: Display the number of lines in a file.
* wc -l example.txt



* **Count Words Only:**
  + Command: wc -w filename
  + Example: Display the number of words in a file.
* wc -w example.txt



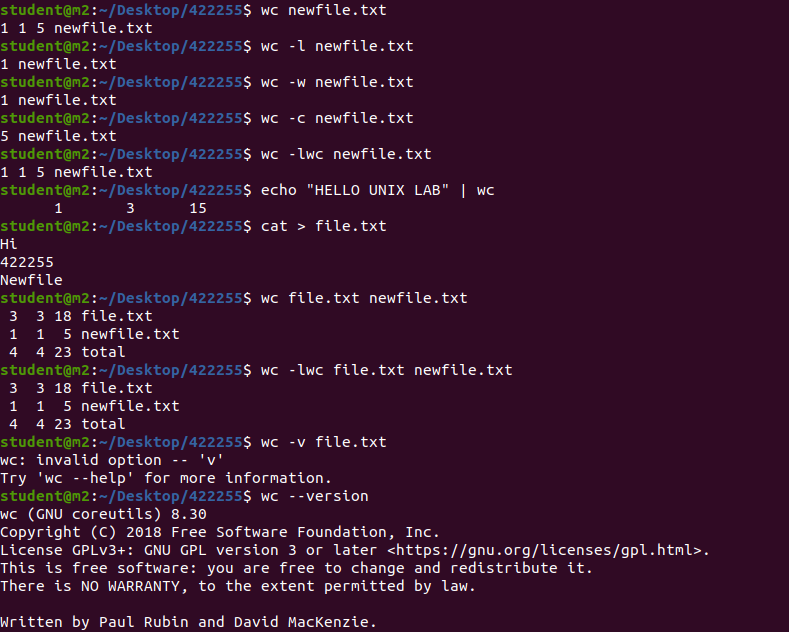
* **Count Characters Only:**
  + Command: wc -c filename
  + Example: Display the number of characters in a file.
* wc -c example.txt
* **Display File Names with Counts:**
  + Command: wc -lwc filename
  + Example: Display the number of lines, words, and characters in a file, along with the file name.
* wc -lwc example.txt
* **Count from Standard Input:**
  + Command: echo "Hello World" | wc
  + Example: Count lines, words, and characters from standard input.  
    bash
* echo "Hello World" | wc
* **Count Multiple Files:**
  + Command: wc file1 file2
  + Example: Count lines, words, and characters for multiple files.  
    bash
* wc file1 file2
* **Total Count for Multiple Files:**
  + Command: wc -lwc file1 file2
  + Example: Display the total number of lines, words, and characters for multiple files.
* wc -lwc file1 file2



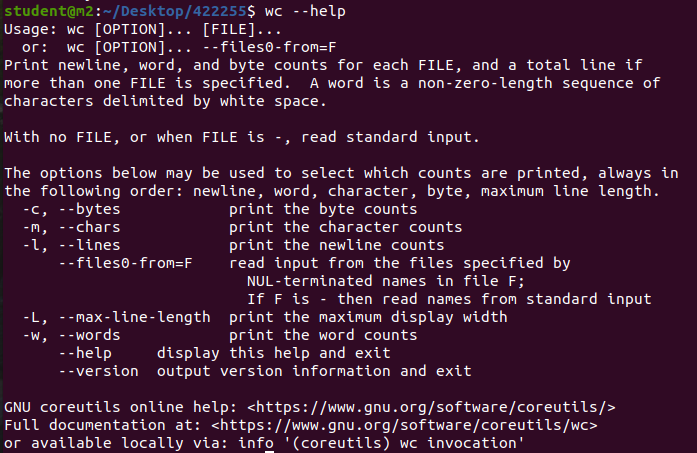
* **Verbose Mode:**
  + Command: wc -v filename
  + Example: Display additional information about the counts.
* wc -v example.txt



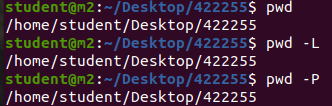
* **Version Information:**
  + Command: wc --version
  + Example: Display version information for wc.
* wc --version

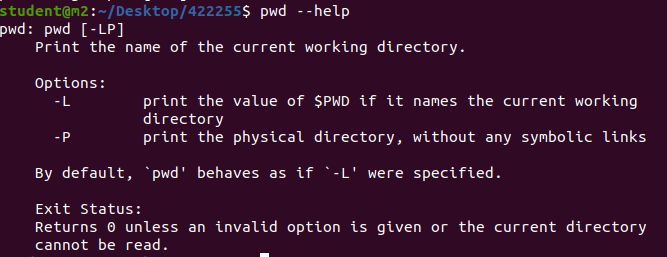


* **Help Information:**
  + Command: wc --help
  + Example: Display help information for wc.  
    bash
* wc --help

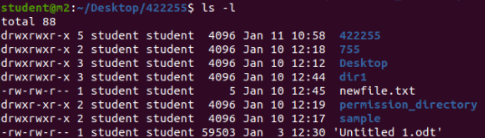
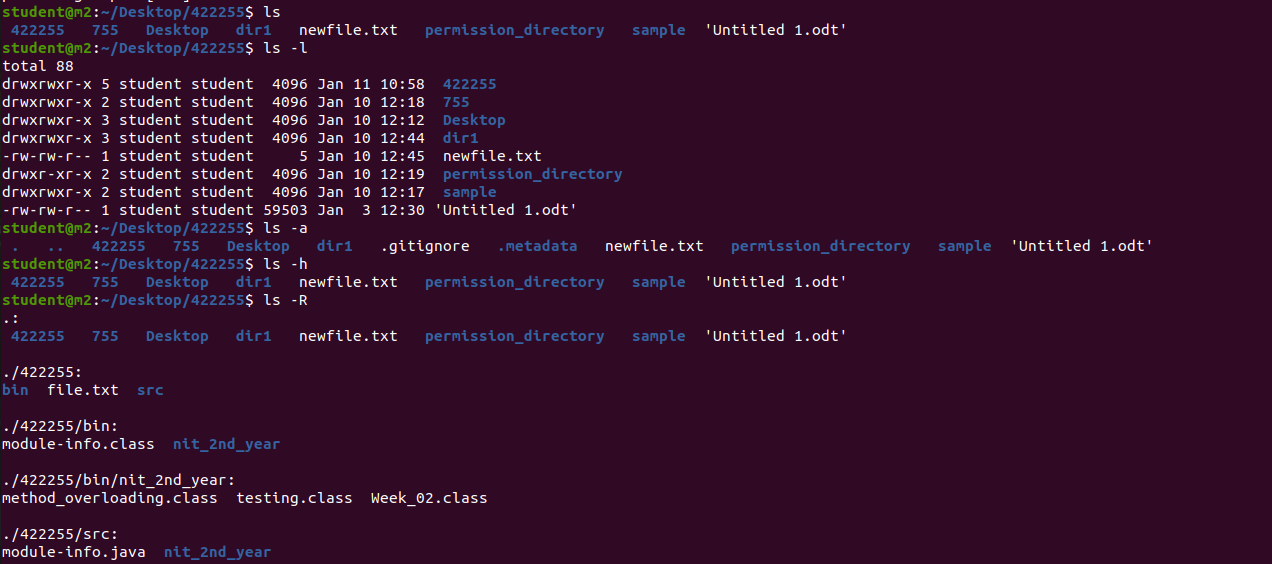


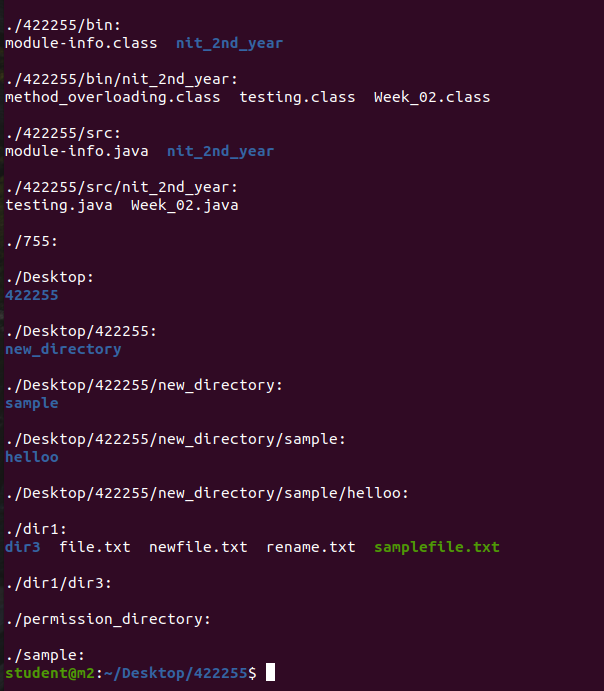
**6. pwd**

* The pwd command is used to print the current working directory. Here are some common options for the pwd command along with examples:
  + **Basic Usage:**
    - Command: pwd
    - Example: Display the current working directory.
* pwd
* 
* **Logical Path -L:**
  + Command: pwd -L
  + Example: Display the logical path of the current working directory (resolved links).
* pwd -L
* 
* **Physical Path -P:**
  + Command: pwd -P
  + Example: Display the physical path of the current working directory (without resolving links).
* pwd -P
* ****
* **Help Information:**
  + Command: pwd --help
  + Example: Display help information for pwd.  
    bash
* pwd --help



**7. ls**

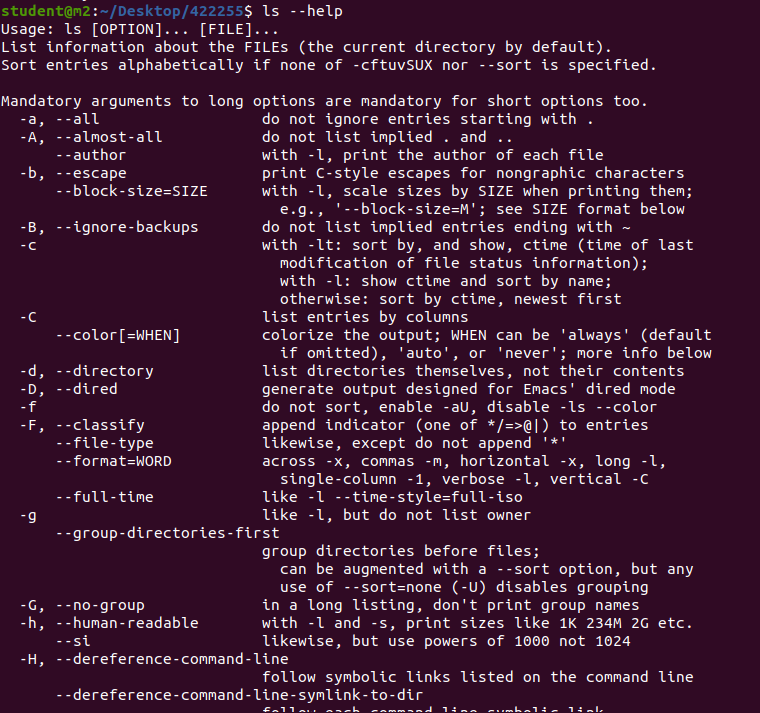
* The ls command is used to list directory contents. Here are some common options for the ls command along with examples:
  + **Basic Usage:**
    - Command: ls
    - Example: List files and directories in the current directory.  
      bash
* ls
* **Long Format -l:**
  + Command: ls -l
  + Example: Display detailed information about files and directories in long format.  
    bash
* ls -l
* 
* **All Files -a:**
  + Command: ls -a
  + Example: List all files, including hidden files.  
    bash
* ls -a
* **Human-Readable Sizes -h:**
  + Command: ls -h
  + Example: Display file sizes in a human-readable format.  
    bash
* ls -h
* **Recursive Listing -R:**
  + Command: ls -R
  + Example: Recursively list all files and directories in the current directory and its subdirectories.
* ls -R



* **Sort by Modification Time -t:**
  + Command: ls -t
  + Example: List files and directories, sorted by modification time.  
    bash
* ls -t
* **Sort by Size -S:**
  + Command: ls -S
  + Example: List files and directories, sorted by size (largest first).  
    bash
* ls -S

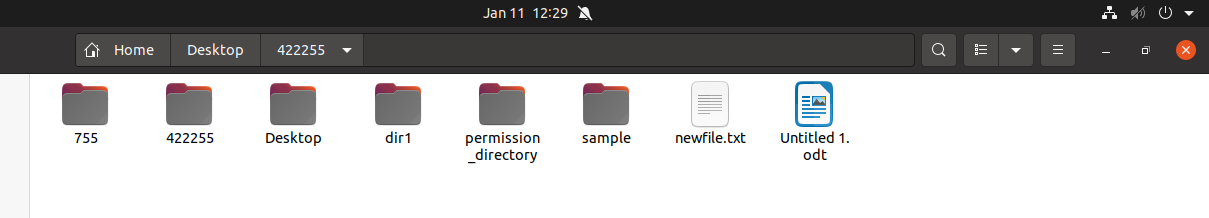


* **Reverse Order -r:**
  + Command: ls -r
  + Example: List files and directories in reverse order.  
    bash
* ls -r
* **Colorized Output --color:**
  + Command: ls --color
  + Example: Display colorized output for better file type visibility.  
    bash
* ls --color
* **List by File Type -F:**
  + Command: ls -F
  + Example: Append a symbol to indicate the type of each entry (e.g., / for directories).  
    bash
* ls -F
* **Version Information:**
  + Command: ls --version
  + Example: Display version information for ls.  
    bash
* ls --version
  + 
* **Help Information:**
  + Command: ls --help
  + Example: Display help information for ls.  
    bash
* ls --help

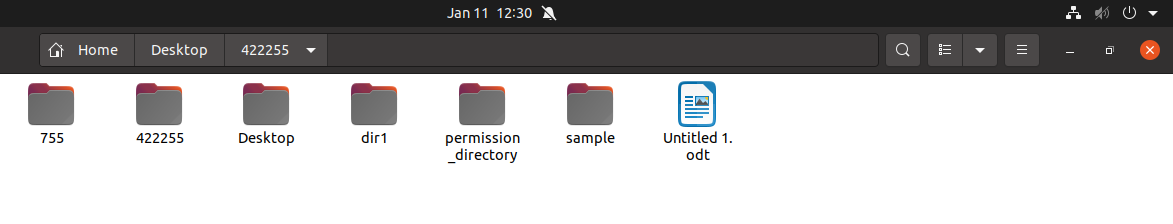
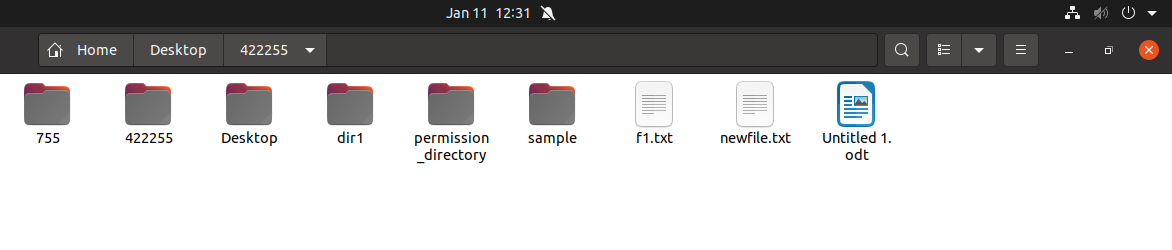


**8. Remove (rm)**

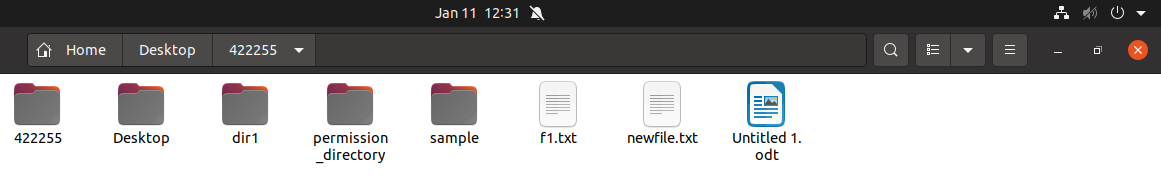
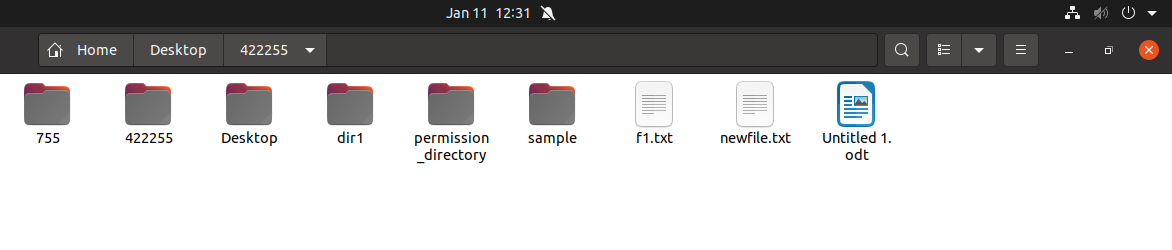
* The rm command in Unix-like operating systems is used to remove or delete files and directories. It has various options to control its behavior. Here are some common options with examples:
  + **Remove a File:**bash
* rm filename



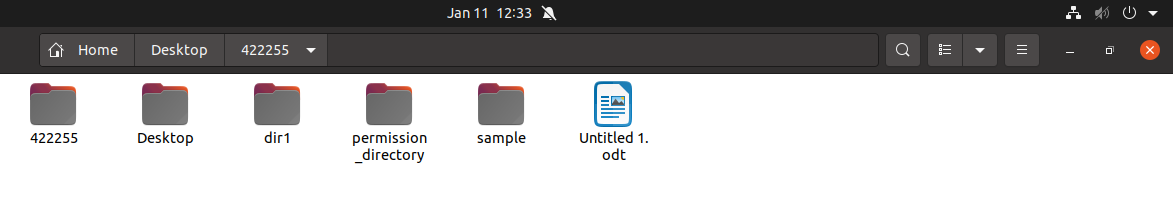
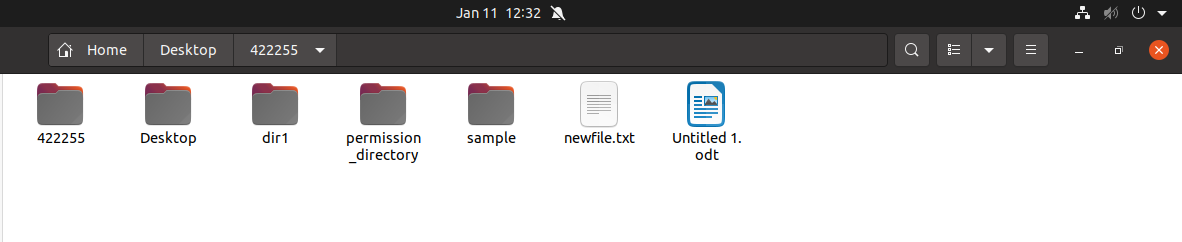
* This command removes the specified file.
* **Remove Multiple Files:**bash
* rm file1 file2 file3



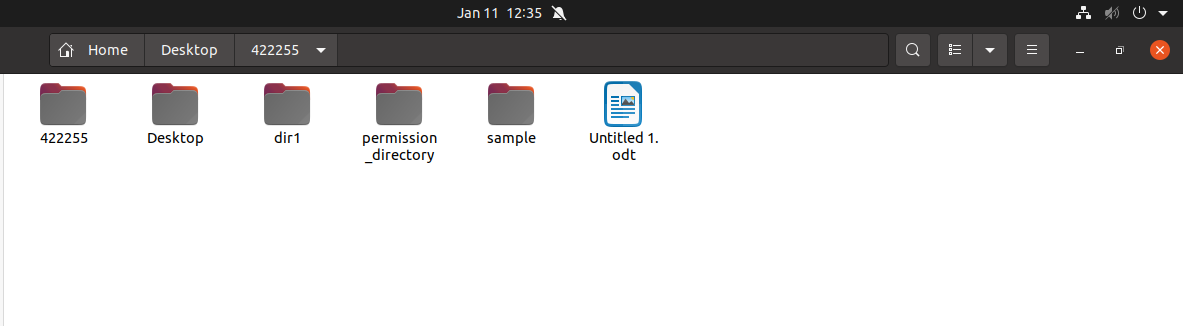
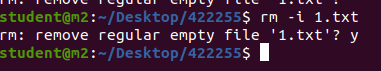
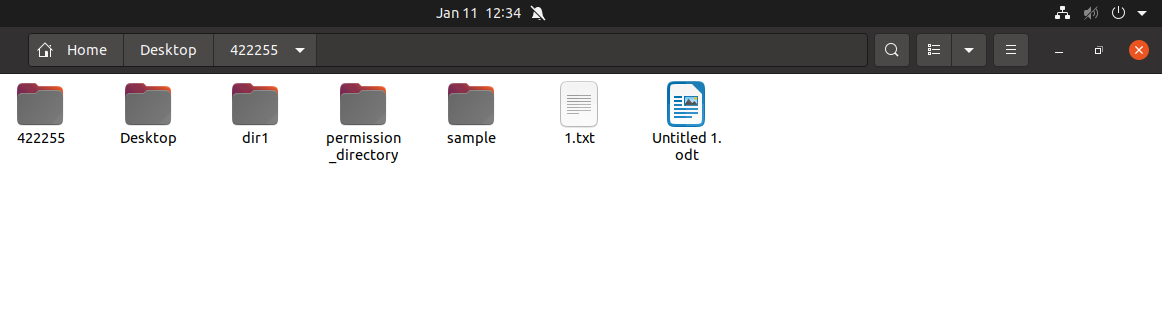
* You can specify multiple files to be removed in a single command.
* **Remove a Directory:**bash
* rm -r directory\_name



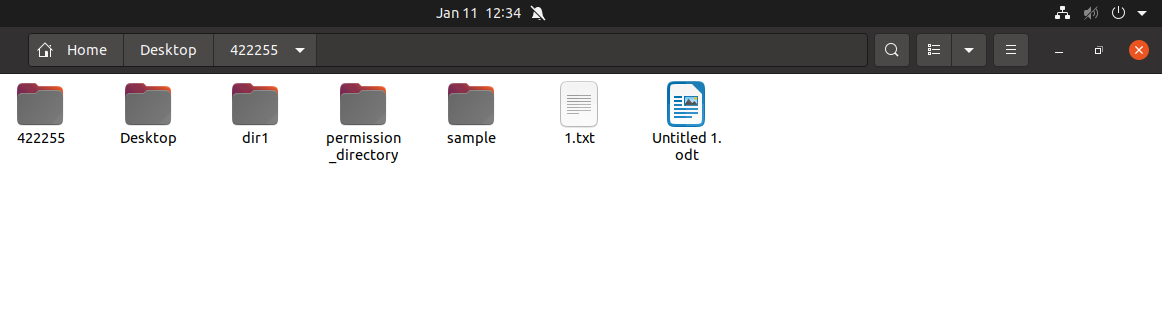
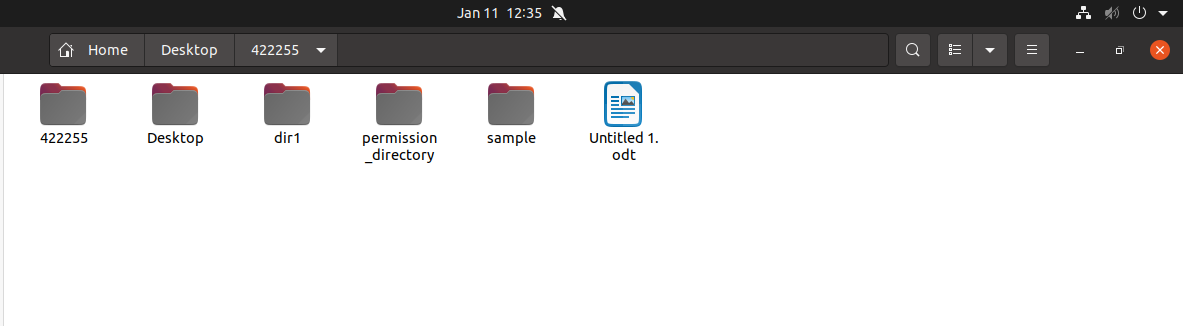
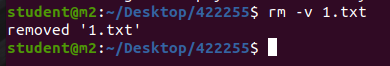
* The -r option is used to remove directories and their contents recursively.
* **Force Removal without Confirmation:**bash
* rm -f filename

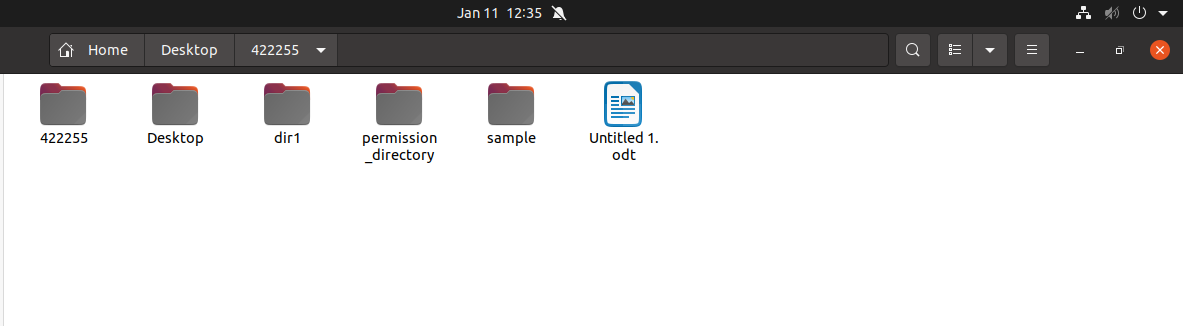
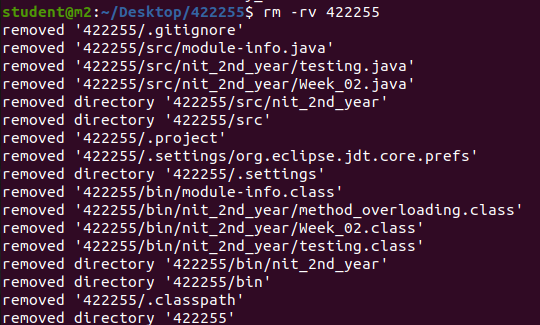


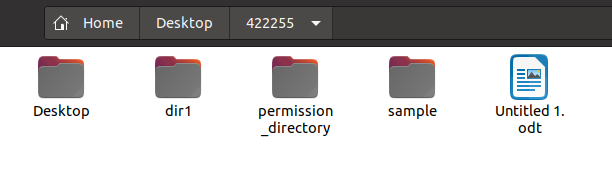
* The -f option forces the removal without prompting for confirmation. Be careful with this option, as it doesn't ask for confirmation before removing files.
* **Interactive Mode (Prompt before Removal):**bash
* rm -i filename



* The -i option makes the rm command interactive, prompting for confirmation before removing each file.
* **Verbose Mode (Display Details):**bash
* rm -v filename

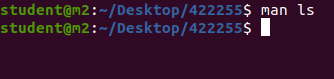


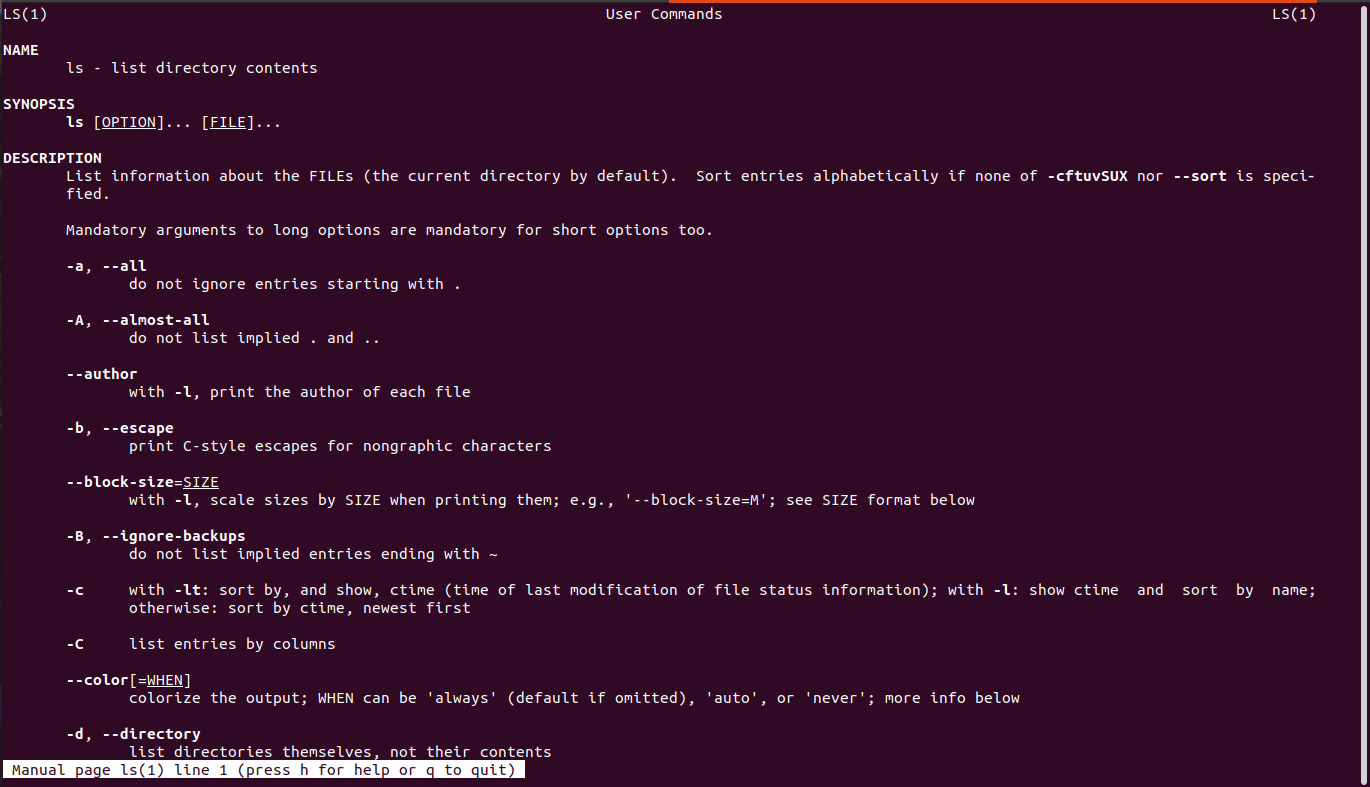
* The -v option makes rm operate in verbose mode, displaying the name of each file as it is removed.
* **Remove Empty Directories:**
* rm -d empty\_directory
* The -d option removes empty directories.
* **Remove a Directory and Its Contents Verbosely:**rm -rv directory\_name



* This removes the directory and its contents recursively, displaying each file's name as it is removed.

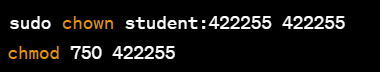
**9.man ls**





* This command will display the manual page for the ls command, which is used to list files and directories. You can replace "ls" with the name of any other command to view its manual page.

**10. CHMOD**

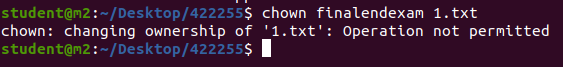
* The chmod command in Unix-like operating systems is used to change the permissions of files and directories. Permissions are assigned to three categories: owner, group, and others. Here are some common options for the chmod command:
* **Symbolic Mode:**
* chmod [ugoa...][[+-=][rwxXstugo...]...][, ...] file\_name
* The symbolic mode allows you to specify permissions symbolically. For example:
  + chmod u+x file\_name adds execute permission for the owner.
  + chmod go-rw file\_name removes read and write permissions for the group and others.
* **Numeric Mode:**
* chmod [ugoa...][[+-=][0-7]...][, ...] file\_name
* The numeric mode allows you to set permissions using octal values. For example:
  + chmod 755 file\_name sets read, write, and execute permissions for the owner, and read and execute permissions for group and others.
* **Recursive Change (Apply to Subdirectories):**
* chmod -R permissions directory\_name
* The -R option is used to change permissions recursively, applying changes to the specified directory and its subdirectories.
* **Change the Permission of a Symbolic Link:**
* chmod -h permissions symbolic\_link
* The -h option changes the permissions of the symbolic link itself, rather than the target it points to.
* **Verbose Mode (Display Changes):**
* chmod -v permissions file\_name
* The -v option makes chmod operate in verbose mode, displaying the changes made.
* **Change the Default ACL:**bash
* chmod +a "user:username:permissions" file\_name
* This command adds an Access Control List (ACL) entry for a specific user.
* **Remove All Permissions for Others:**bash
* chmod o= file\_name
* 
* This removes all permissions for others (those not in the owner or group).
* **Set the Setuid, Setgid, and Sticky Bits:**bash
* chmod u+s file\_name



* This sets the setuid bit, which allows a program to run with the privileges of the file's owner.  
  bash  
  chmod g+s directory\_name
* This sets the setgid bit on a directory, making files created in the directory inherit the group ownership of the directory.  
  bash  
  chmod +t directory\_name
* This sets the sticky bit on a directory, allowing only the owner of a file to delete or rename it in the directory.
* The chown command in Unix-like operating systems is used to change the user and/or group ownership of files and directories. Here are some common options for the chown command:

**11. Change User and Group Ownership:**

* chown new\_owner:new\_group file\_name
* This command changes the user and group ownership of the specified file or directory to the specified owner and group.
* **Change User Ownership Only:**bash
* chown new\_owner file\_name



* This command changes only the user ownership of the specified file or directory.
* **Change Group Ownership Only:**bash
* chown :new\_group file\_name
* This command changes only the group ownership of the specified file or directory.
* **Recursive Change (Apply to Subdirectories):**bash
* chown -R new\_owner:new\_group directory\_name
* The -R option is used to change ownership recursively, applying changes to the specified directory and its subdirectories.
* **Preserve Root Ownership:**bash
* chown --preserve-root new\_owner:new\_group /
* This option prevents chown from modifying the ownership of the root directory (/).
* **Reference Ownership from Another File:**bash
* chown --reference=reference\_file target\_file
* This option sets the ownership of the target file to be the same as that of the reference file.
* **Verbose Mode (Display Changes):**bash
* chown -v new\_owner:new\_group file\_name
* The -v option makes chown operate in verbose mode, displaying the changes made.
* **Only Report Changes (No Action):**bash
* chown -c new\_owner:new\_group file\_name
* The -c option reports only when a change is made, without performing any actual changes.
* **Reference Ownership from Symbolic Link:**
* chown --dereference new\_owner:new\_group symbolic\_link
* This option follows symbolic links and changes the ownership of the target file
* The chgrp command in Unix-like operating systems is used to change the group ownership of files or directories. Here are some common options for the chgrp command:

**12. Change Group Ownership**

* chgrp group\_name file\_name
* This command changes the group ownership of the specified file or directory to the specified group.
* **Recursive Change (Apply to Subdirectories):**
* chgrp -R group\_name directory\_name
* The -R option is used to change the group ownership recursively, applying changes to the specified directory and its subdirectories.
* **Reference Group from Another File:**
* chgrp --reference=reference\_file target\_file
* This option sets the group ownership of the target file to be the same as that of the reference file.
* **Preserve Root Ownership:**
* chgrp --preserve-root group\_name /
* This option prevents chgrp from modifying the ownership of the root directory (/).
* **Verbose Mode (Display Changes):**
* chgrp -v group\_name file\_name
* The -v option makes chgrp operate in verbose mode, displaying the changes made.
* **Only Report Changes (No Action):**
* chgrp -c group\_name file\_name
* The -c option reports only when a change is made, without performing any actual changes.
* **Reference Group from Symbolic Link:**
* chgrp --dereference group\_name symbolic\_link
* This option follows symbolic links and changes the group ownership of the target file.

**2. Implement for loop, while loop, if statement, arithmetic, logical, conditional expressions using bc and echo commands**

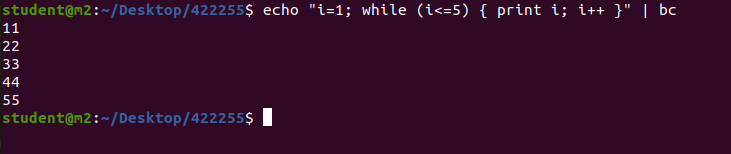
* **For Loop:**
* bash
* echo "for (i=1; i<=5; i++) { print i }" | bc



* This will print numbers from 1 to 5.

### **While Loop:**

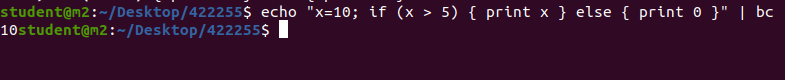
* bash
* echo "i=1; while (i<=5) { print i; i++ }" | bc



* This will also print numbers from 1 to 5 using a while loop.

### **If Statement:**

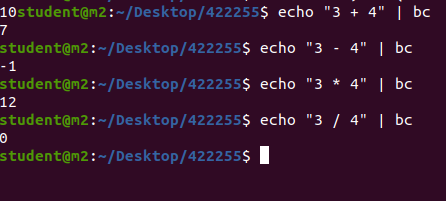
* bash
* echo "x=10; if (x > 5) { print x } else { print 0 }" | bc



* This will print 10 since the condition (x > 5) is true.

### **Arithmetic Expressions:**

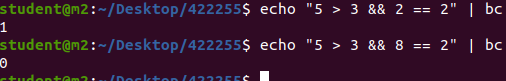
* bash
* echo "3 + 4" | bc



* This will output the result of the arithmetic expression 3 + 4.

### **Logical Expressions:**

* echo "5 > 3 && 2 == 2" | bc



* This will output 1 (true) since both conditions are true.

### **Conditional Expression:**

* echo "x=5; y=3; x > y ? x : y" | bc



* This will output 5, as it evaluates the expression x > y ? x : y and returns the value of x because the condition is true.