MEGHNAD SAHA INSTITUTE OF TECHNOLOGY

*Techno Complex, Madurdaha,Beside NRI Complex, Post-Uchhepota, Kolkata 700 150*

LABORATORY NOTE BOOK

MAKAUT EVEN SEMESTER 2023



[BACHELOR OF COMPUTER APPLICATION]

[UNIX AND SHELL PROGRAMMING LAB (BCAC601)]

[RUPAK SARKAR]

ROLL NO: 31001221010 REGN. NO.: 213101001210001

STREAM: BCA SEMESTER: VI (6TH)

YEAR: 3RD YearSESSION: 2021-2024



MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY



MEGHNAD SAHA INSTITUTE OF TECHNOLOGY

*Techno Complex,. Madurdaha,Beside NRI Complex, Post-Uchhepota, Kolkata 700 150*

“LIST OF ASSIGNMENT/EXPERIMENT SUBMISSION DETAILS”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL.**  **NO.** | **ASSIGNMENT / EXPERIMENT NAME** | **PLANNED DATE OF SUBMISSION** | **ACTUAL DATE OF SUBMISION** | **CHECKED BY** | **REMARKS (ANY DEVIATION REGARDING SUBMISSION DATES, CONTENT, FORMAT, ETC)** |
| 1. | Displaying pathname of current directory. | 18/03/2024 | 18/03/2024 |  |  |
| 2. | How to change Directory. | 18/03/2024 | 18/03/2024 |  |  |
| 3. | How to make new directory. | 18/03/2024 | 18/03/2024 |  |  |
| 4. | How to remove directory. | 18/03/2024 | 18/03/2024 |  |  |
| 5. | Listing contents of directory (Is and its options). | 18/03/2024 | 18/03/2024 |  |  |
| 6. | Absolute pathname, Relative pathname. | 18/03/2024 | 18/03/2024 |  |  |
| 7. | Using dot (.) and dotdot (..) | 18/03/2024 | 18/03/2024 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| OBSERVATIONS / COMMENTS ON THE OVERALL PERFORMANCE : |

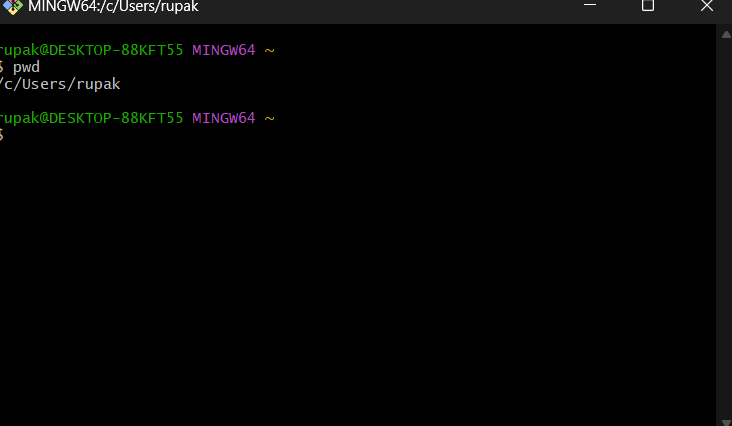
Signature in full with date Signature in full with date

**Faculty / Technical Assistant Lab Examiner**

**Q.1 Display Current Working Directory (PWD).**

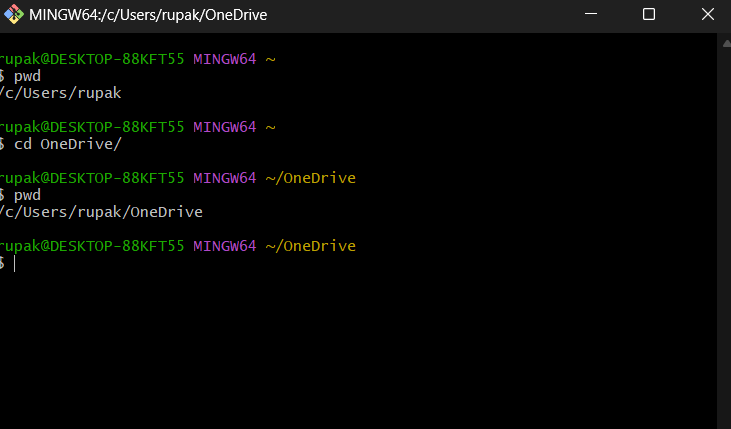
PWD – It is also called Present Working Directory. This command shows the present working directory of the user.

Syntax - pwd



**Q.2. How to change Directory?**

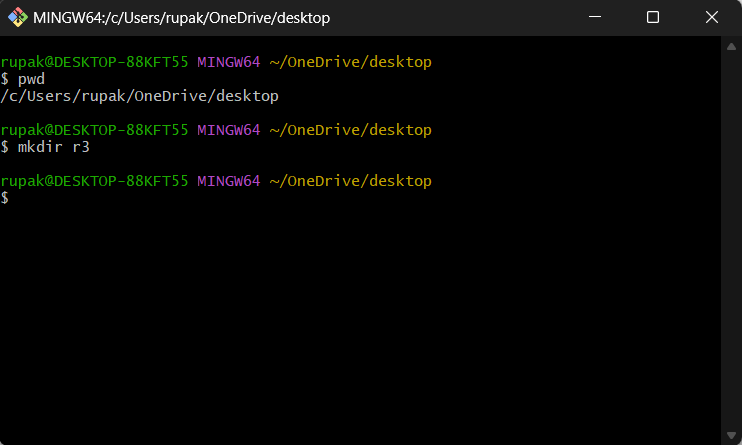
To change directory, we need to use the “cd” command. The “cd” command allows the users to change their working directory.



**Q.3. How to make a new directory.**

To make a new directory, we need to use the “mkdir” command. The “mkdir” command allows us to create a new directory on our present working directory.

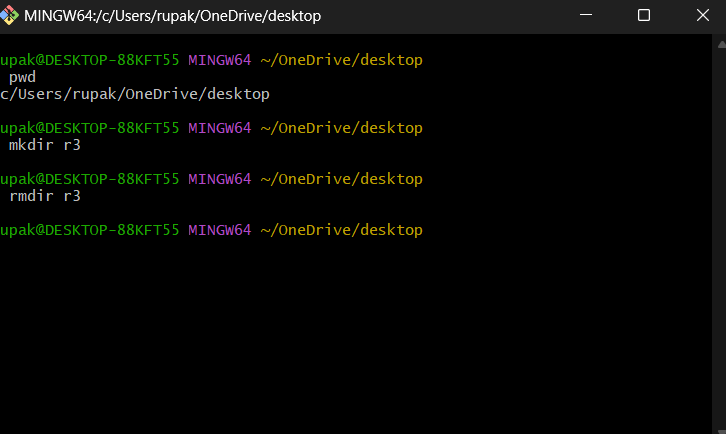
Syntax – mkdir [directory\_name]



**Q.4. How to remove and existing directory.**

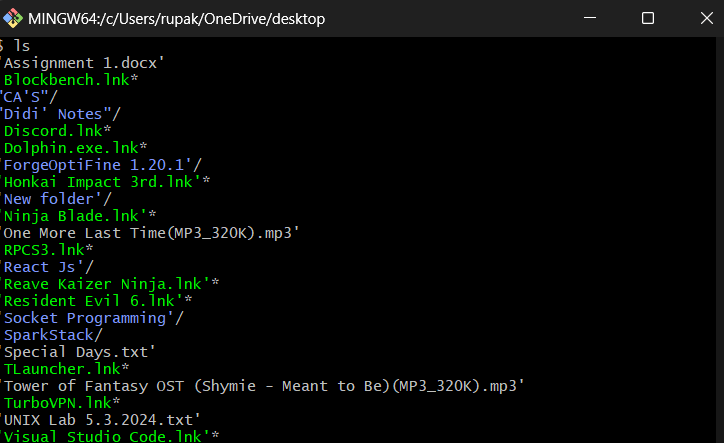
To remove an existing directory, we need to use the “rmdir” command. The “rmdir” command allows us to remove and already existing directory.

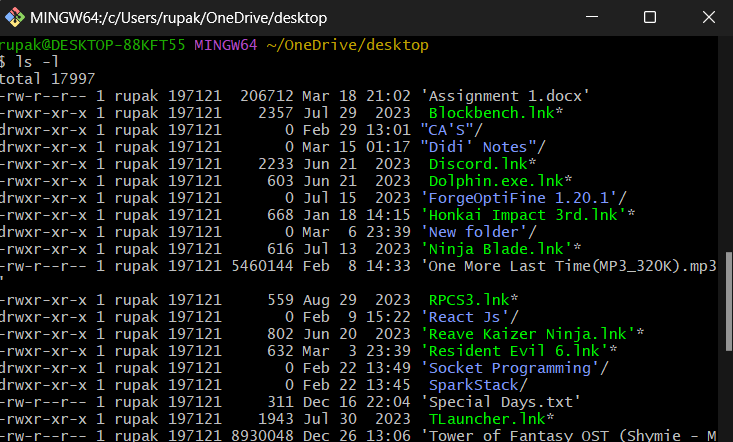
Syntax – rmdir [existing\_directory]

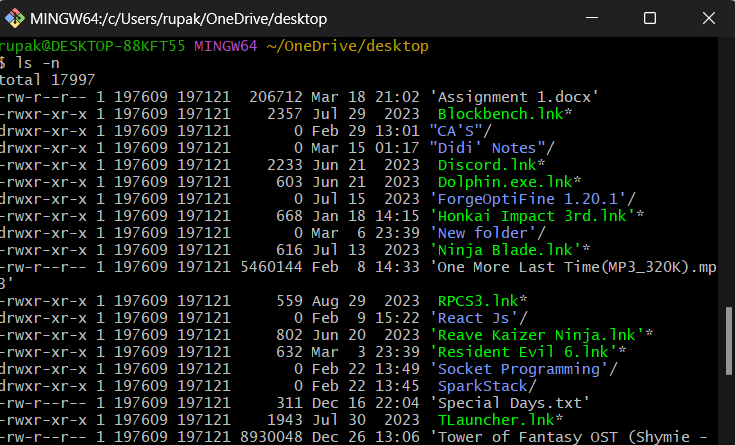


**Q.5. Listing commands.**

The “ls” command allows us to list all the files and directories.







**Q.6. Absolute Pathname, Relative Pathname.**

**Absolute Pathname –**

In Unix-like operating systems, an absolute pathname refers to the full path of a file or directory starting from the root directory (/). It specifies the complete location of the file or directory within the file system hierarchy. Absolute pathnames are independent of the current working directory and always start from the root directory.

Here's an example of an absolute pathname:

```

/home/user/documents/example.txt

```

In this example:

- `/` represents the root directory.

- `home` is a directory within the root directory.

- `user` is a directory within the `home` directory.

- `documents` is a directory within the `user` directory.

- `example.txt` is a file within the `documents` directory.

Absolute pathnames are useful for referencing files and directories in scripts, commands, and configuration files, as they provide an unambiguous reference to the location of a file or directory regardless of the current working directory.

**Absolute Pathname –**

In Unix-like operating systems, a relative pathname specifies the location of a file or directory relative to the current working directory. Unlike absolute pathnames, relative pathnames do not start from the root directory (/). Instead, they indicate the path relative to the current directory.

Here's an example of a relative pathname:

```

../../documents/example.txt

```

In this example:

- `..` represents the parent directory.

- `../../` means go up two levels from the current directory.

- `documents` is a directory located two levels up from the current directory.

- `example.txt` is a file within the `documents` directory.

Relative pathnames are handy when navigating within the file system hierarchy from a specific starting point. They are often used in scripts, commands, and when referencing files and directories within a specific context. However, they may become ambiguous if used in different locations or contexts, as they depend on the current working directory.

**Q.7. Using dot (.) dotdot (..).**

**Dot (.) –**

In Unix-like operating systems, the `.` (dot) represents the current directory. It is commonly used in file system navigation and path manipulation.

Here are some common uses of `.` in Unix:

1. Current Directory Reference: When you use `.` in a pathname, it refers to the directory you are currently in. For example, if your current directory is `/home/user`, then `./example.txt` refers to a file named `example.txt` in the current directory.

2. Execute Commands: In Unix shell scripting, `./` is often used to explicitly specify that a script or executable should be executed from the current directory. For instance, `./script.sh` would run a shell script named `script.sh` in the current directory.

3. Relative Pathname: `./` can be used as part of a relative pathname to specify a file or directory in the current directory. For example, `./subdirectory/file.txt` refers to a file named `file.txt` located within a subdirectory of the current directory.

4. File Redirection: In Unix shells, `./` can be used in conjunction with input/output redirection. For example, `./script.sh > output.txt` would run `script.sh` from the current directory and redirect its output to a file named `output.txt`.

Overall, `.` is a fundamental concept in Unix-like systems, providing a concise way to reference the current directory in various contexts.

**Dot-Dot (..) –**

In Unix-like operating systems, `..` (dot-dot) represents the parent directory of the current directory. It is commonly used in file system navigation and path manipulation.

Here are some common uses of `..` in Unix:

1. Parent Directory Reference: When you use `..` in a pathname, it refers to the directory that contains the current directory. For example, if your current directory is `/home/user/documents`, then `../example.txt` refers to a file named `example.txt` in the parent directory of `/home/user/documents`.

2. Relative Pathname: `..` can be used as part of a relative pathname to specify a file or directory in the parent directory. For example, `../other\_directory/file.txt` refers to a file named `file.txt` located within a directory named `other\_directory` in the parent directory of the current directory.

3. Directory Traversal: `..` can be used for directory traversal, allowing you to move up the directory hierarchy. For example, if you are in a deeply nested directory structure, you can use `cd ..` to move up one level to the parent directory.

4. Symbolic Links: `..` is also used in symbolic links. For instance, if you create a symbolic link named `link` in a directory and point it to `..`, accessing `link` would refer to the parent directory.

Overall, `..` is a fundamental concept in Unix-like systems, providing a concise way to reference the parent directory in various contexts, especially when navigating through the file system.