A white background with black text

Description automatically generated

A screen shot of numbers and symbols

Description automatically generated



A group of green and blue symbols

Description automatically generated with medium confidence



A black screen with green and white text

Description automatically generated



A black background with letters and numbers

Description automatically generated



A computer screen with green and white text

Description automatically generated



A black screen with green and white text

Description automatically generated



A screen shot of a computer

Description automatically generated



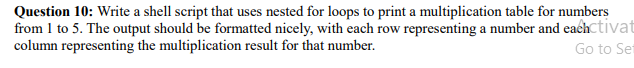
A black background with different colored letters

Description automatically generated with medium confidence



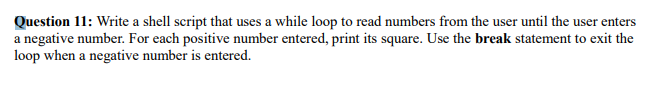
A screenshot of a computer program

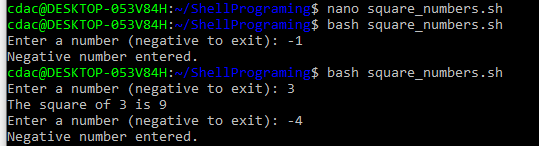
Description automatically generated



A screen shot of a computer

Description automatically generated





A close-up of a paper

Description automatically generated

Prints Hello, World! to the terminal.

* name="Productive"

Assigns the value Productive to the variable name.

* touch file.txt

Creates an empty file named file.txt if it does not exist or updates its modification timestamp if it does.

* 1s -a

Lists all files and directories in the current directory, including hidden ones (those starting with a dot).

* rm file.txt

Deletes the file named file.txt.

* cp filel.txt file2.txt

Copies the contents of file1.txt to a new file named file2.txt.

mv file.txt /path/to/ directory/

* chmod 755 script. Sh

Changes the permissions of script.sh to 755 (read, write, and execute for the owner; read and execute for group and others).

* grep "pattern" file.txt

Searches for lines containing pattern in file.txt and prints those lines.

* kill PID

Sends a termination signal to the process with the specified process ID (PID).

* mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt

\* Creates a directory named mydir.

\* Changes into that directory.

\* Creates an empty file named file.txt.

\* Writes Hello, World! into file.txt.

\* Displays the contents of file.txt.

* 1s -1 | grep ".txt"

Lists files and directories with detailed information and filters the results to show only those containing. txt in their names.

* cat file1.txt file2.txt | sort | uniq

\* Concatenates the contents of file1.txt and file2.txt.

\* Sorts the combined contents.

\* Removes duplicate lines.

* 1s -1 | grep "^d"

Lists files and directories with detailed information and filters the results to show only directories (entries starting with d).

* grep -r "pattern" /path/to/ directory/

Recursively searches for pattern in all files under / path/to/directory/.

* cat filel.txt file2.txt | sort | uniq -d

\* Concatenates the contents of file1.txt and file2.txt.

\* Sorts the combined contents.

\* Displays only the duplicate lines.

* chmod 644 file.txt

Changes the permissions of file.txt to 644 (read and write for the owner; read-only for group and others).

* cp -r source\_directory destination\_directory

Recursively copies the contents of source\_directory to destination directory.

* 19. find /path/to/search -name"\*. txt"

Searches for files with a. txt extension in /path/to/ search directory.

* chmod u+x file.txt

Adds execute permissions for the user (owner) of file.txt.

* echo SPATH

Displays the current value of the PATH environment variable, which lists directories that are searched for executable files.

A black and white text

Description automatically generated

1. ls is used to list files and directories in a directory.

True - ls is used to list files and directories in a directory.

2. mv is used to move files and directories.

True - mv is used to move files and directories.

1. cd is used to copy files and directories.

False - cd is used to change the current directory, not copy files and directories. The command cp is used to copy files and directories.

1. pwd stands for "print working directory" and displays the current directory.

True - pwd stands for "print working directory" and displays the current directory.

5. grep is used to search for patterns in files.

True - grep is used to search for patterns in files.

1. chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.

True - chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to the group and others.

1. mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist.

True - mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1, and if directory1 does not exist, it will create it as well.

1. rm -rf file.txt deletes a file forcefully without confirmation.

True - rm -rf file.txt deletes a file forcefully without confirmation. The -r option is typically used for directories, but -f forces deletion without prompts, and applying it to a file removes it without asking.

Identify the Incorrect Commands:

1. chmodx is used to change file permissions.

Incorrect - chmodx is not a valid command. The correct command for changing file permissions is chmod.

2. cpy is used to copy files and directories.

Incorrect - cpy is not a valid command. The correct command for copying files and directories is cp.

3. mkfile is used to create a new file.

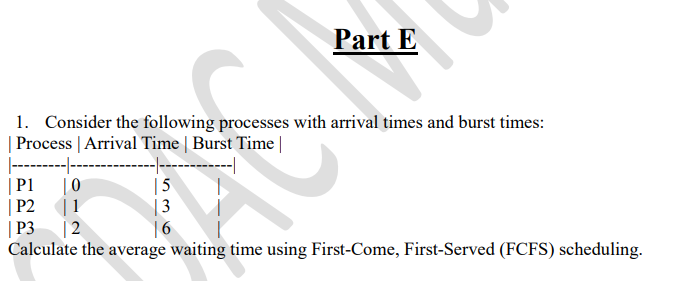
\*Incorrect\* - mkfile is not a standard command for creating files in Unix-like systems. Typically, touch is used to create a new file, or echo/cat can be used with redirection.

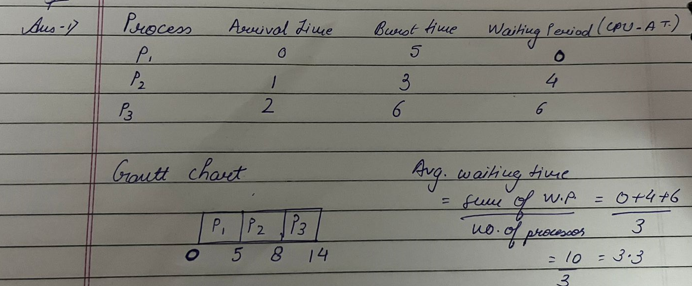
4. catx is used to concatenate files.

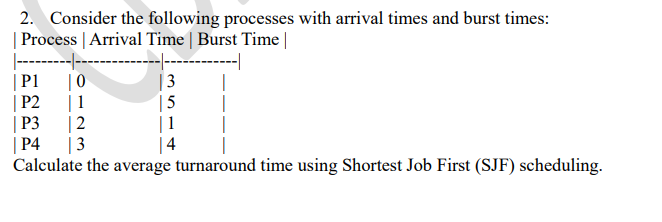
Incorrect - catx is not a valid command. The correct command for concatenating and displaying file content is cat.

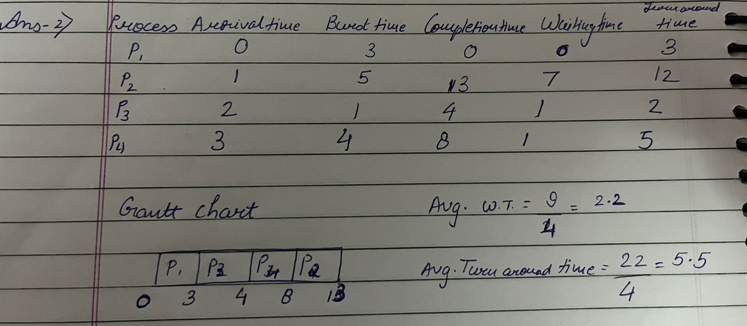
5. rn is used to rename files.

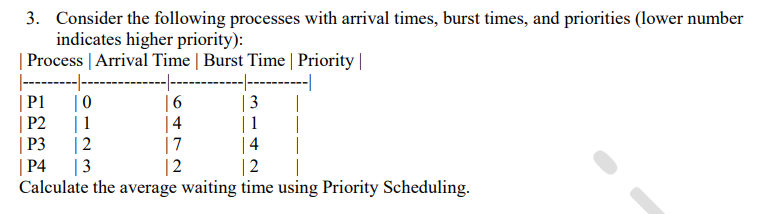
Incorrect - rn is not a valid command. The correct command to rename (or move) files is mv.

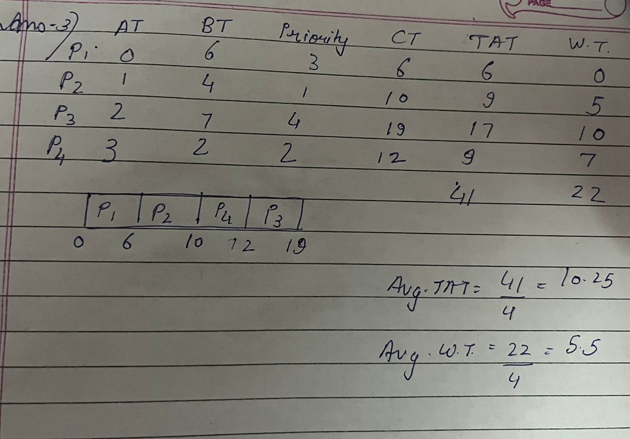


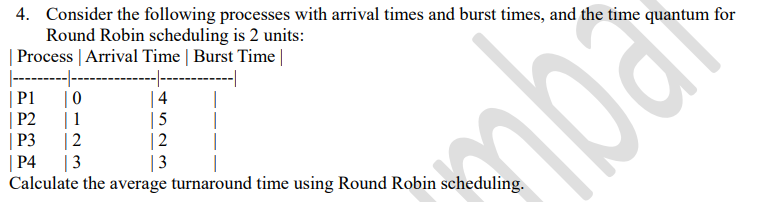


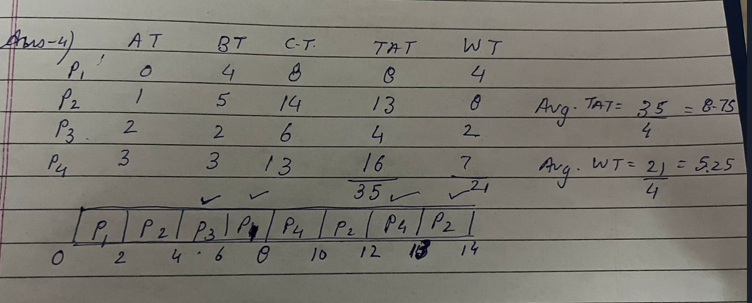












5. Consider a program that uses the fork() system call to create a child process. Initially, the parent process has a variable x with a value of 5. After forking, both the parent and child processes increment the value of x by 1. What will be the final values of x in the parent and child processes after the fork() call?

