

# Linux + C Programming Cheat Sheet

## 1. File Creation & Editing

Command:  
vi filename.c

Example Question:  
"Write a C program named frequency.c to implement parent-child communication using pipes."

vi Basics:  
- i → Insert mode (start typing code).  
- Esc → Back to command mode.  
- :wq → Save & quit.  
- :q! → Quit without saving.

## 2. Compiling a C Program

Command:  
gcc filename.c -o outputname

Example Question:  
"Compile your frequency.c program into an executable called frequency."

gcc frequency.c -o frequency

## 3. Running the Executable

Command:  
./outputname

Example Question:  
"Run your compiled frequency program and enter 5 wavelengths as input."

./frequency

## 4. Checking Running Processes

Command:  
ps

Example Question:  
"After running your program that uses fork(), check the child processes created."

## 5. Killing a Process (if it hangs)

Command:  
kill -9 PID

Replace PID with the process ID from ps.

## 6. Compiling with Debugging

Command:

```
gcc -Wall filename.c -o outputname
```

Example Question:

"Debug and recompile your program if you get warnings about unused variables.

## 7. Viewing File Contents

Commands:

```
cat filename.c
```

```
less filename.c
```

Example Question:

"Check if your name and IT number are included in the C file comments.

## 8. Clearing the Terminal

Command:

```
clear
```

Example Question:

"Clear the screen before re-running your program to avoid confusion.

## 9. Running Multiple Commands

Command:

```
gcc frequency.c -o frequency && ./frequency
```

Example Question:

"Compile and execute your program in a single command.

## 10. File Management

Commands:

ls → List files in directory.

rm filename → Remove file.

cp old new → Copy file.

Example Question:

"List the files in your directory to confirm that frequency.c and frequency exist.

## Sample C Program (frequency.c)

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>

int main() {
    int pipe1[2], pipe2[2];
```

```

float lambda[5], freq[5];
float v = 1000.0;

// Create pipes
pipe(pipe1);
pipe(pipe2);

if (fork() == 0) {
    // Child-1 process
    close(pipe1[1]); // Close write end of pipe1
    close(pipe2[0]); // Close read end of pipe2

    // Read wavelengths from parent
    read(pipe1[0], lambda, sizeof(lambda));

    // Calculate frequencies
    for (int i = 0; i < 5; i++) {
        freq[i] = v / lambda[i];
    }

    // Write frequencies to pipe2
    write(pipe2[1], freq, sizeof(freq));

    if (fork() == 0) {
        // Child-2 process
        close(pipe2[1]); // Close write end of pipe2
        read(pipe2[0], freq, sizeof(freq));

        printf("Frequencies calculated from Child-2:\n");
        for (int i = 0; i < 5; i++) {
            printf("f[%d] = %.2f Hz\n", i+1, freq[i]);
        }
        exit(0);
    }
    wait(NULL); // Wait for Child-2
    exit(0);
} else {
    // Parent process
    close(pipe1[0]); // Close read end of pipe1

    printf("Enter 5 wavelength values (in meters):\n");
    for (int i = 0; i < 5; i++) {
        scanf("%f", &lambda[i]);
    }

    // Write wavelengths to pipe1
    write(pipe1[1], lambda, sizeof(lambda));

    wait(NULL); // Wait for Child-1
}

return 0;
}

```