**🛠️ Shell Command Cheat Sheet – System & Library Functions**

| **Function / Macro** | **Description** | **Returns / Behavior** |
| --- | --- | --- |
| getcwd(char \*buf, size\_t size) | Gets current working directory and stores it in buf. | Pointer to buffer on success, nullptr on failure |
| chdir(const char \*path) | Changes the current working directory. | 0 on success, -1 on error |
| getenv(const char \*name) | Gets the value of an environment variable. | Pointer to value, or nullptr if not found |
| perror(const char \*msg) | Prints error message based on errno with custom prefix. | No return value (prints to stderr) |
| fork() | Creates a new child process. | 0 in child, PID in parent, -1 on failure |
| execvp(const char \*file, char \*const argv[]) | Replaces current process with a new program. | No return on success, -1 on failure |
| exit(int status) | Terminates the current process with an exit code. | Ends process execution |
| waitpid(pid\_t pid, int \*status, int options) | Waits for child process to terminate. | PID of child, or -1 on failure |

**🧩 C++ Standard Library Functions & Utilities**

| **Function / Object** | **Description** | **Notes** |
| --- | --- | --- |
| getline(cin, input) | Reads entire line from input stream into a string. | Handles user input |
| stoi(string) | Converts string to integer. | Throws invalid\_argument or out\_of\_range |
| find(begin, end, value) | Searches for a value in a container (e.g., vector). | Returns iterator or .end() |
| isspace(char) | Checks if a character is whitespace. | Used for command tokenization |
| const\_cast<char\*> | Converts const char\* to char\*. | Needed for execvp() args |

**📁 File I/O (Streams)**

| **Object / Function** | **Description** | **Used for** |
| --- | --- | --- |
| ifstream file(filename) | Reads from a file. | Loads history |
| getline(file, line) | Reads a line from an input file stream. | Reads each command from history |
| ofstream file(filename) | Writes to a file. | Saves history |
| file << string << endl | Writes a string followed by a newline. | Writes history entries line by line |

**🎨 Terminal Color Codes (ANSI Escape Sequences)**

| **Escape Code** | **Effect** |
| --- | --- |
| \033[0m | Reset color |
| \033[31m | Red |
| \033[32m | Green |
| \033[34m | Blue |
| \033[1m | Bold text |