1. Retrieve the filename from command line arguments:

- Get the third element of the `process.argv` array, which represents the filename to be watched.

const filename = process.argv[2];

2. Check if a filename is provided:

- Ensure that a filename is provided as a command line argument. If not, throw an error.

if (!filename) {

throw Error('A file to watch must be specified!');

}

3. Extract the command and its arguments from command line arguments:

- Extract the command from `process.argv[3]` and additional arguments from `process.argv.slice(4)`.

const command = process.argv[3];

const commandArgs = process.argv.slice(4);

1. **process.argv:** This is an array in Node.js that contains the command line arguments passed to the script. The first two elements of this array are:
   * **process.argv[0]**: The path to the Node.js executable.
   * **process.argv[1]**: The path to the script being run.
2. **.slice(4):** This is a method applied to the **process.argv** array. The **slice** method is used to extract a portion of the array starting from the specified index. In this case, **slice(4)** means starting from the fifth element of the array (index 4).

So, **process.argv.slice(4)** is taking all the elements of **process.argv** starting from the fifth element (index 4) onward. It effectively extracts all the command line arguments passed to the script after the script name and the Node.js executable path.

4. Watch for file changes and handle events:

- Define an event handler for file changes and renames. If either event occurs, spawn a child process using the specified command and its arguments, including the filename being watched.

const watchHandler = (eventType, filename) => {

if (eventType === 'change' || eventType === 'rename') {

const ls = spawn(command, [...commandArgs, filename]);

ls.stdout.pipe(process.stdout);

}

};

5. Set up the file watcher:

- Establish a file watcher on the specified filename, using the `watchHandler` function to handle events.

fs.watch(filename, watchHandler);

6. Print a message indicating the file is being watched:

- Display a message in the console, indicating that the script is now watching the specified file for changes.

console.log(`Now watching ${filename} for changes...`);