bashAI User Manual

Introduction

`bashAI` is a custom-built Unix-like shell designed to enhance command-line interactions with features like:

- Built-in command support
- Piping and redirection
- Background task execution
- Command aliasing
- Environment variable management
- Command history tracking

This manual will guide you through the usage of all major functionalities available in `bashAI`.

Starting the Shell

To start the `bashAI` shell, open your terminal and run:

```
python3 main.py
```

You'll see a custom prompt indicating you're now in the `bashAI` shell environment.

---

1. Built-in Commands

`cd`

Change the current working directory.

```
cd /path/to/directory
```

`echo`

Display a line of text.

```
echo Hello, World!
```

`export`

Set environment variables.

```
export VAR_NAME=value
```

To use the variable:

```
echo $VAR_NAME
```

`history`

Display the list of previously entered commands.

```
history
```

`alias`

Create a shortcut for a command.

```
alias ll='ls -l'
```

Use `unalias` to remove an alias:

```
unalias ll
```

---

2. External Commands

Any external executable available on your system's `$PATH` can be used:

```
ls -la
cat file.txt
python3 script.py
```

---

3. Piping (|)

You can chain commands by piping output of one into another:

```
cat file.txt | grep "search_term"
```

---

4. Redirection

Output Redirection

- Overwrite output to a file:

```
echo "Hello" > output.txt
```

- Append output to a file:

```
echo "Again" >> output.txt
```

Input Redirection

- Read input from a file:

```
wc -l < file.txt
```

---

5. Background Execution (&)

To run a command in the background:

```
sleep 10 &
```

You’ll see a message with the process ID.

---

6. Quitting the Shell

To exit the shell, use:

```
exit
```

Or press `Ctrl + D`.

---

7. Error Handling

`bashAI` handles:

- Unrecognized commands
- Missing files or directories
- Syntax errors in piping/redirection

You will see a helpful error message instead of a crash.

Example Session

```
> cd ~/Documents
> echo "Test" > file.txt
> cat file.txt | grep "Test"
> export NAME=Sujal
> echo $NAME
> sleep 5 &
> history
> exit
```

Notes

- Aliases and variables are session-specific and not persistent across restarts.
- You can extend functionality by modifying the codebase in `executer.py` and `parser_self.py`.