
SynapseOS – User Manual

Version 1.0 – Developed by: Group-7

1. Introduction

SynapseOS is a custom-built operating system designed from scratch using low-level system programming (C, Assembly, and a custom kernel).

It features a modern graphical interface, interactive desktop environment, built-in applications, and early-stage AI capabilities.

This manual provides instructions for installation, usage, OS features, system behavior, and troubleshooting.

2. System Requirements

SynapseOS is lightweight and works even on limited hardware.

Minimum Requirements

Component	Requirement
CPU	x86 (32-bit)
RAM	64 MB
Storage	100 MB
Virtualization	VirtualBox or QEMU recommended

Recommended Setup

- Processor: Dual-Core
 - RAM: 256 MB+
 - Display: 1024×768
-

3. Installing SynapseOS

3.1 Installation in VirtualBox

1. Open VirtualBox → Create New VM
2. Choose:
 - **Type:** Other
 - **Version:** Other/Unknown (32-bit)
3. Assign at least **256 MB RAM**
4. Add the SynapseOS .iso
5. Start the VM

3.2 Booting

On boot:

- Bootloader loads
 - Kernel starts
 - Graphics mode initializes
 - Desktop GUI appears
-

4. Desktop Environment

SynapseOS features a clean, custom-built GUI.

4.1 Components

- **Top Taskbar** – contains system title “SynapseOS”
 - **Clock** – shows real-time
 - **App Launcher Icons** – Notepad, AI Shell
 - **Wallpaper** – customizable background
 - **Mouse Support** – full GUI interaction
-

5. Applications

5.1 Notepad Application

A lightweight text editor built into SynapseOS.

Features

- Write and edit text
- Supports keyboard input
- Supports text selection and caret movements
- Save text (internal memory for now)
- Close using:
 - **Esc key**
 - Or clicking the close control (if enabled)

Keyboard Shortcuts

Shortcut	Action
Ctrl + S	Save text
Esc	Close Notepad
Arrows	Move caret

Enter	New line
Backspace	Delete

6. Smart Shell (AI Command System)

SynapseOS supports a Smart Shell that can interpret natural-language typed commands.

6.1 Example Commands

User Command	System Action
“open notepad”	Launches Notepad
“close notepad”	Closes Notepad
“show time”	Displays clock
“clear screen”	Clears shell screen
“help”	Shows command list

7. Planned AI Integration

(Documentation included for academic submission)

7.1 AI Assistant Overview

A built-in AI assistant is planned, with following features:

Core Features

- Voice Recognition (using offline/on-device STT)
- Natural-language understanding
- System automation:
 - Opening/closing apps
 - Writing in Notepad automatically
 - Controlling OS windows
- General Chatbot powered by OpenAI API
 - Uses your API key (secure storage planned)

8. Using AI Assistant (When Implemented)

Example Voice Commands

Voice Command	Action
“Write an email in Notepad”	AI opens Notepad and types text
“Explain quantum computing”	AI chatbot responds

“Close all windows”

OS performs action

9. File System

Currently minimal and memory-based.

Capabilities

- In-memory storage for Notepad text
 - Future support for FAT/EXT filesystem planned
-

10. Troubleshooting

Issue: Notepad not closing

Solution: Press ESC key.

Issue: Screen frozen

Solution: Restart VM.

Issue: Smart Shell not responding

Solution:

- Re-open shell
- Ensure keyboard focus

Issue: Build errors

Refer to kernel logs and ensure GCC cross-compiler installed.
