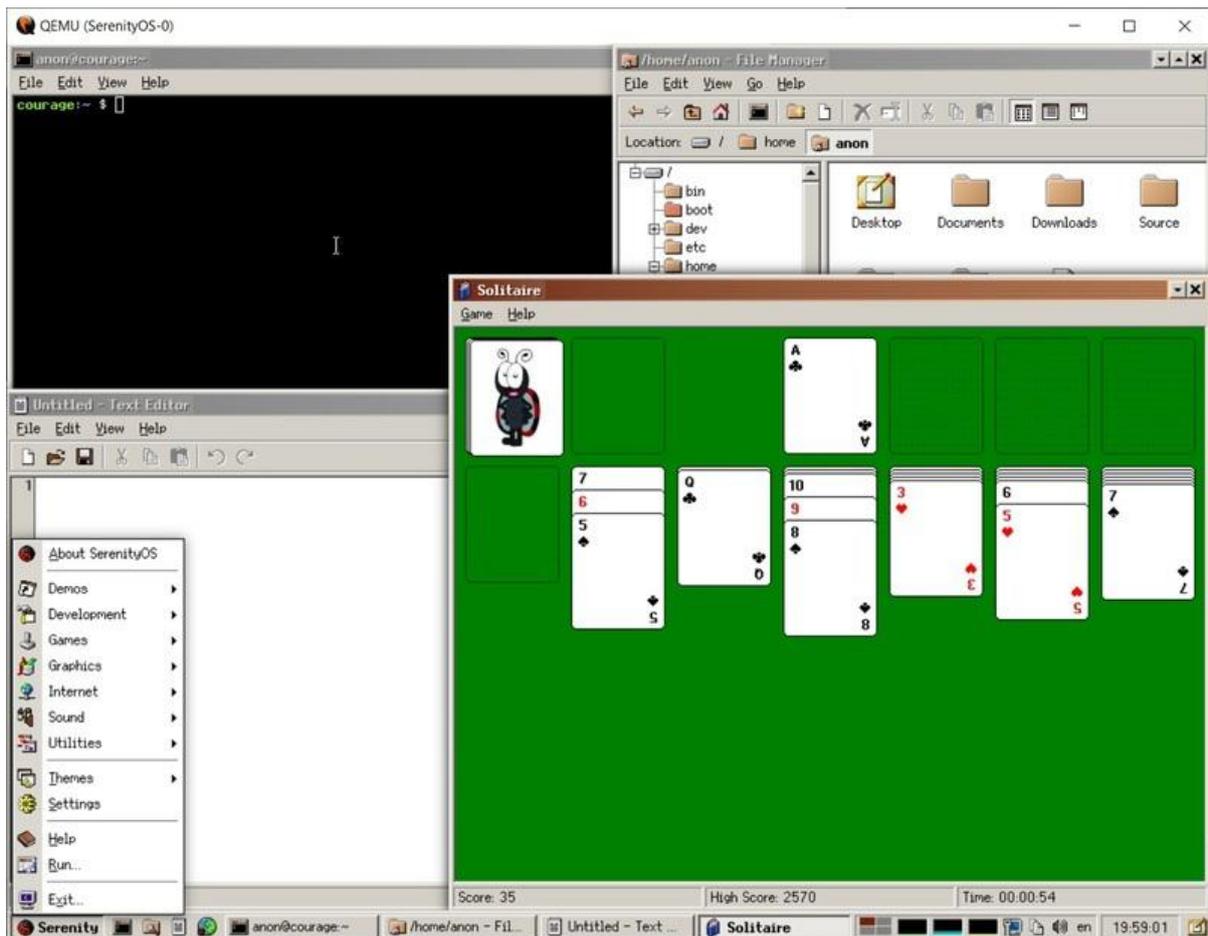




SerenityOS - The dream of the '90s is alive!

Created by Matt G



<https://learn.adafruit.com/serenityos-build-and-run-keep-the-90s-dream-alive>

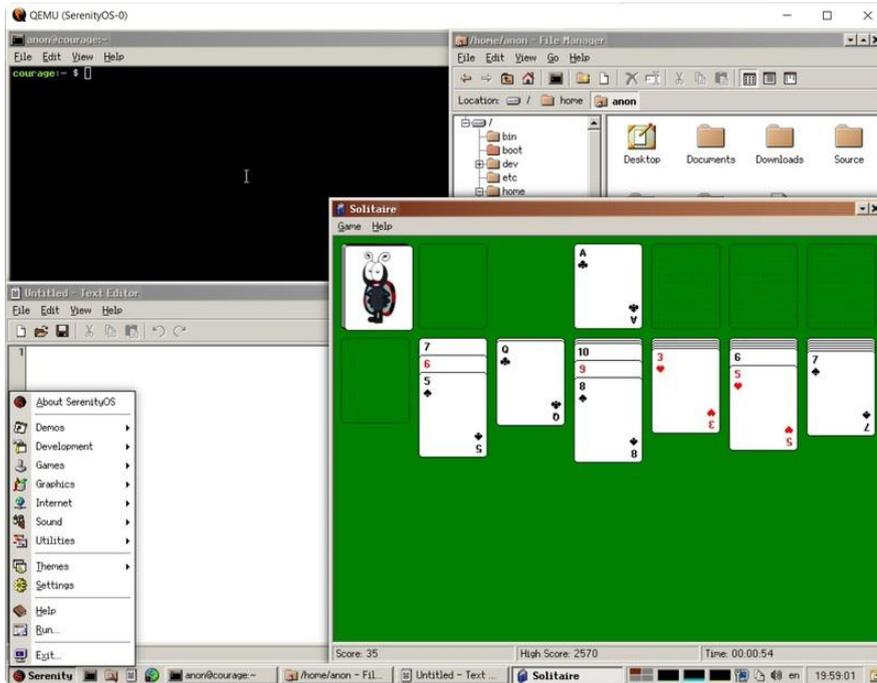
Last updated on 2023-08-29 04:48:35 PM EDT

Table of Contents

What is SerenityOS?	3
Build (Windows)	4
Build (macOS, Linux)	6
Running	8
More Info / Contributing	12

What is SerenityOS?

SerenityOS is a new operating system written from scratch by Andreas Kling and many, many more open-source contributors. It is described as "a love letter to '90s user interfaces with a custom Unix-like core."



Why the 90s? Probably just feeling nostalgic, but it was an exciting time!

It was the era of Windows 95, Mac System 7, and this new thing called Linux. GUIs had been around for a while but they were improving quickly. It sounds quaint now but we were excited about having more than 16 colors on the screen at once. It was amazing to be able to play a video on your computer. The average web page now probably has more multimedia playing than a typical 1990s computer could handle. The internet was becoming more widely available but unless you were lucky you had to use dial-up to connect.

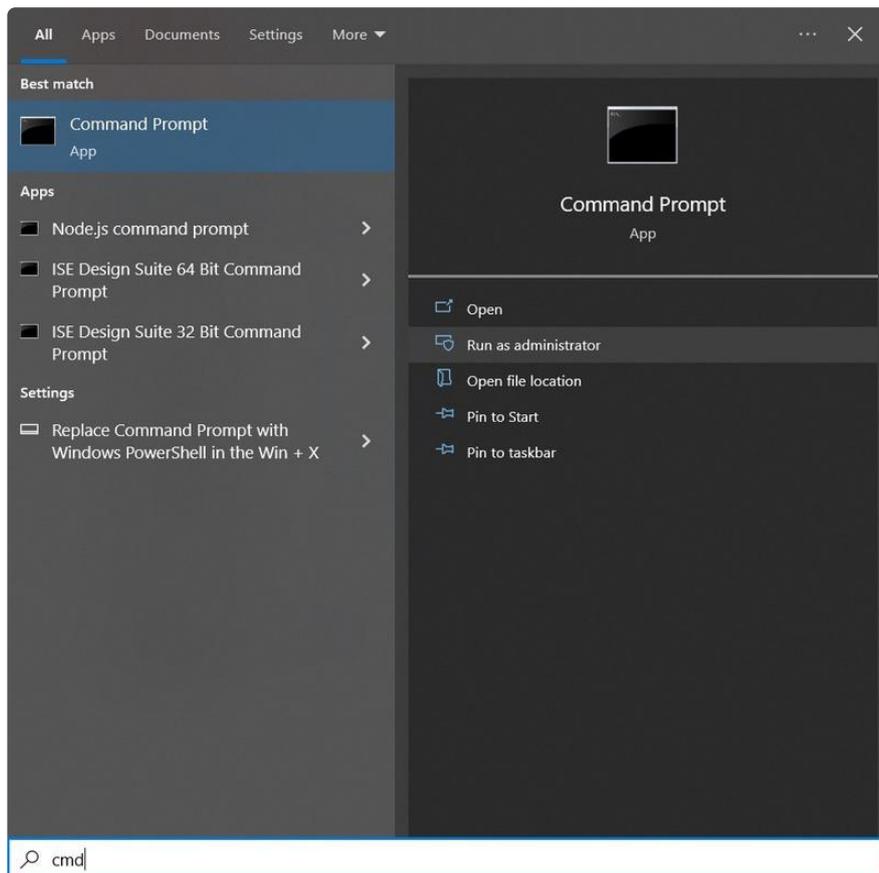
If you want the true 90s experience you should download this at 28.8 kbps and install it from floppy disks. Fortunately for us, SerenityOS only looks like the 90s, it's a fully modern system underneath!

Let's build it and take a look.

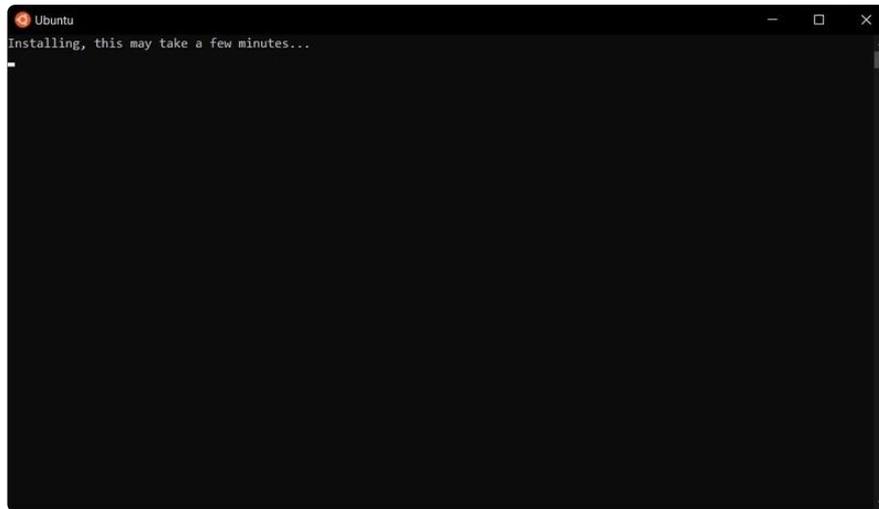
Build (Windows)

In order to run SerenityOS under Windows, you'll need to use the Windows Subsystem for Linux (WSL) version 2, so you need Windows 10 (v2004 or higher) or Windows 11. WSL lets you run Linux binaries in Windows, so you can have a full Linux distribution available without needing a virtual machine or another computer.

To install it you need an administrator command prompt - you can get to one by going to start, typing cmd, then selecting Run as administrator in the options for Command Prompt.



Now type `wsl --install` to start the install process. The default distribution is Ubuntu, but you can change it if you like. Check the list of available distributions with `wsl --list --online` and then install the one you want with `wsl --install -d <Distro Name>`. It can take a while to install - at some point it will tell you to reboot and then finish up, prompting you to create a user account.



Now that you have WSL2, you can start a terminal with the new Ubuntu (or other distro) shortcut in the start menu. Time to build!

First you'll need QEMU for Windows, you can get it at: <https://www.qemu.org/download/#windows> (). Follow the link for the 64-bit installer and get the qemu-w64-setup-YYYYMMDD.exe file. Install at least the Tools and x86_64 System emulation.

Back in WSL2 land we can install the prerequisites to build SerenityOS.

```
sudo apt install build-essential cmake curl libmpfr-dev libmpc-dev  
libgmp-dev e2fsprogs ninja-build qemu-system-gui qemu-system-x86  
qemu-utils ccache rsync unzip texinfo
```

Next we need GCC 11. The default Ubuntu install in WSL2 doesn't have it, so we can add it using these steps:

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test  
sudo apt update  
sudo apt install gcc-11 g++-11
```

SerenityOS uses QEMU version 5 and will build it on its own but it needs a few libraries:

```
sudo apt install libgtk-3-dev libpixman-1-dev libsdl2-dev libspice-  
server-dev
```

All set! Now we can grab the SerenityOS repository.

```
git clone https://github.com/SerenityOS/serenity.git ()
```

```
cd serenity
```

And now build QEMU:

```
Toolchain/BuildQemu.sh
```

```
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Removing outdated /home/drakar/serenity/Build/lagom/TZDB for version 2022a
-- Downloading file tzdb.tar.gz from https://data.iana.org/time-zones/releases/tzdata2022a.tar.gz
-- Extracting TZDB africa from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Extracting TZDB antarctica from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Extracting TZDB asia from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Extracting TZDB australasia from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Extracting TZDB backward from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Extracting TZDB etcetera from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Extracting TZDB europe from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Extracting TZDB northamerica from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Extracting TZDB southamerica from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Extracting TZDB zone1970.tab from /home/drakar/serenity/Build/lagom/TZDB/tzdb.tar.gz...
-- Configuring done
-- Generating done
-- Build files have been written to: /home/drakar/serenity/Build/lagom
[5/15] Performing build step for 'lagom'
[0/2] Re-checking globbed directories...
[32/101] Building CXX object CMakeFiles/LagomCore.dir/home...enity/Userland/Libraries/LibCore/FilePermissionsMask.cpp.o
```

Next build the toolchain:

```
Meta/serenity.sh rebuild-toolchain
```

And finally SerenityOS itself!

```
Meta/serenity.sh run
```

Build (macOS, Linux)

On macOS you'll need [Homebrew \(\)](#) and Xcode installed (get it in the App Store), plus you need to run Xcode once to let it install more tools.

First you need to tell the system to use Xcode's tools:

```
sudo xcode-select --switch /Applications/Xcode.app
```

Now install the prerequisites:

```
brew install coreutils e2fsprogs qemu bash gcc@11 imagemagick ninja
cmake ccache rsync zstd libvirt
```

If you have an Apple Silicon Mac make sure that you only have these tools in one brew environment, arm or x86. Using the native arm tools is going to be faster.

There are two options for handling Serenity's ext2 filesystem, we'll use the simpler way with genext2fs. You can also use macFUSE, but it's a bit more complicated, check out the [build instructions \(\)](#) to do that.

```
brew install genext2fs
```

That's it for prereqs! Now we can start building.

Grab the SerenityOS repository:

```
git clone https://github.com/SerenityOS/serenity.git
```

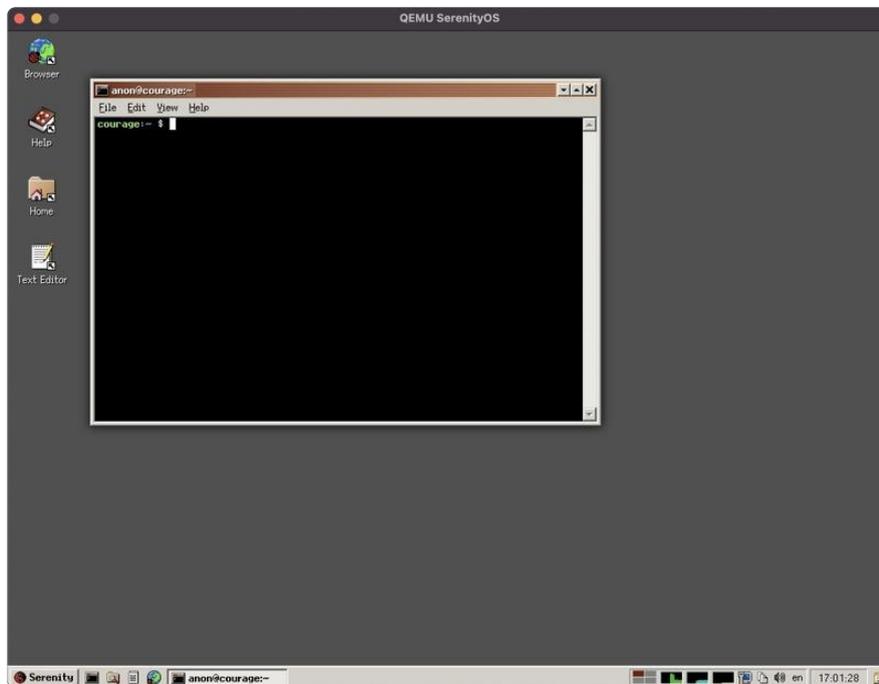
```
cd serenity
```

First build the toolchain:

```
Meta/serenity.sh rebuild-toolchain
```

Now the final step:

```
Meta/serenity.sh run
```



If you're running Linux, you can just install a few libraries and then start building, check out the [build instructions \(\)](#) for your specific distribution if it's not here.

Ubuntu / Debian

```
sudo apt install build-essential cmake curl libmpfr-dev libmpc-dev
libgmp-dev e2fsprogs ninja-build qemu-system-gui qemu-system-x86
qemu-utils ccache rsync unzip texinfo
```

You may need to install GCC 11 as well:

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
sudo apt update
sudo apt install gcc-11 g++-11
```

And some prerequisites for QEMU:

```
sudo apt install libgtk-3-dev libpixman-1-dev libsdl2-dev libspice-
server-dev
```

Arch

```
sudo pacman -S --needed base-devel cmake curl mpfr libmpc gmp
e2fsprogs ninja qemu qemu-arch-extra ccache rsync unzip
```

Now it's the same as above to build the toolchain and run the OS:

```
git clone https://github.com/SerenityOS/serenity.git
```

```
cd serenity
```

Depending on the distribution you may need to build qemu:

```
Toolchain/BuildQemu.sh
```

Then finish up:

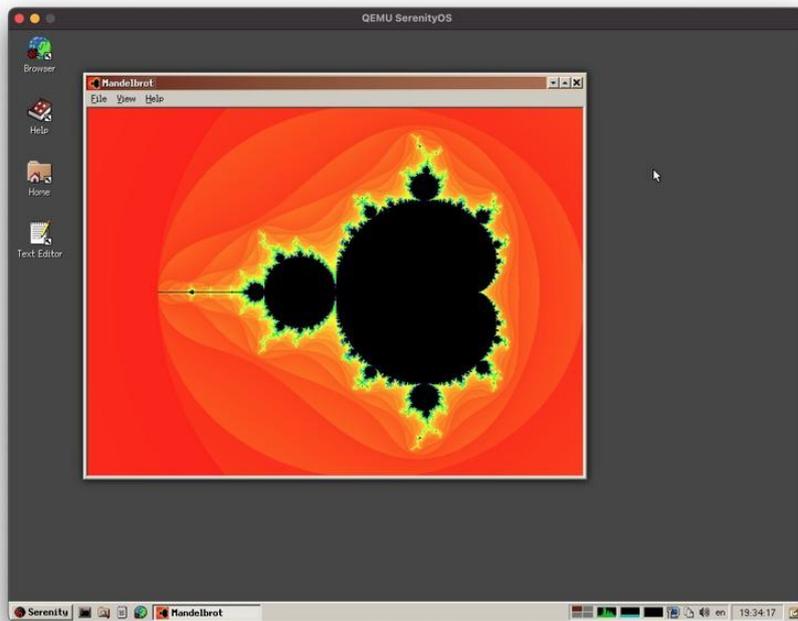
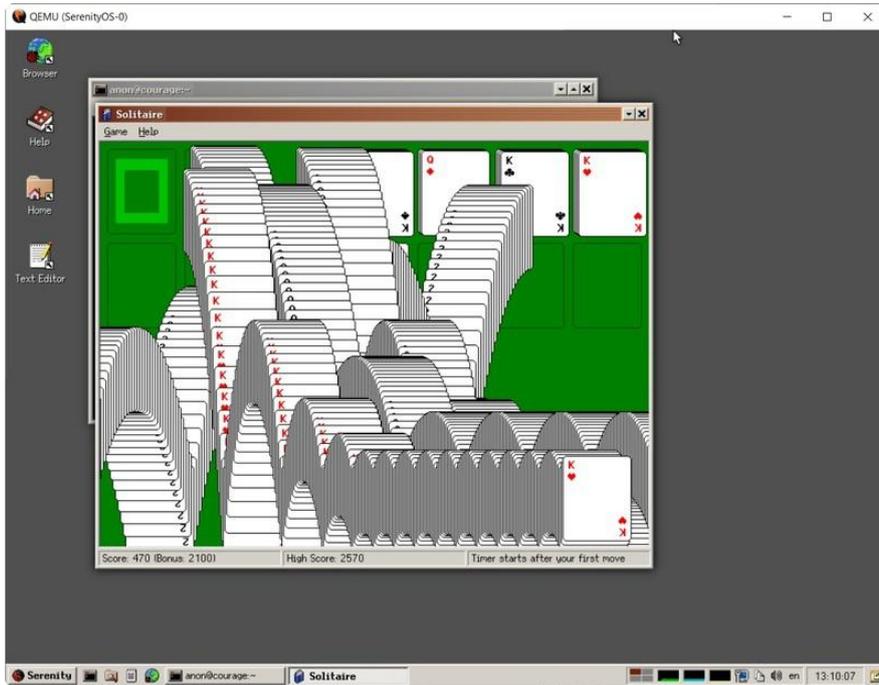
```
Meta/serenity.sh rebuild-toolchain
```

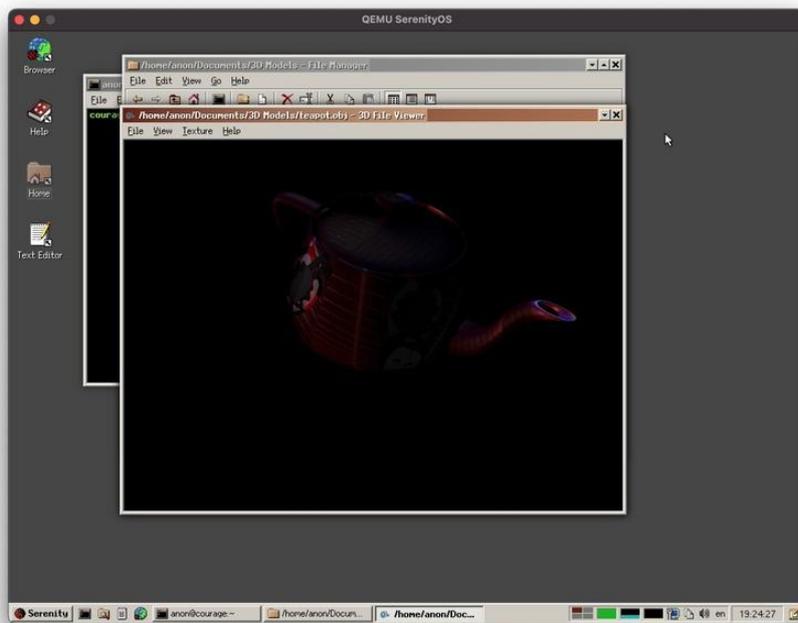
```
Meta/serenity.sh run
```

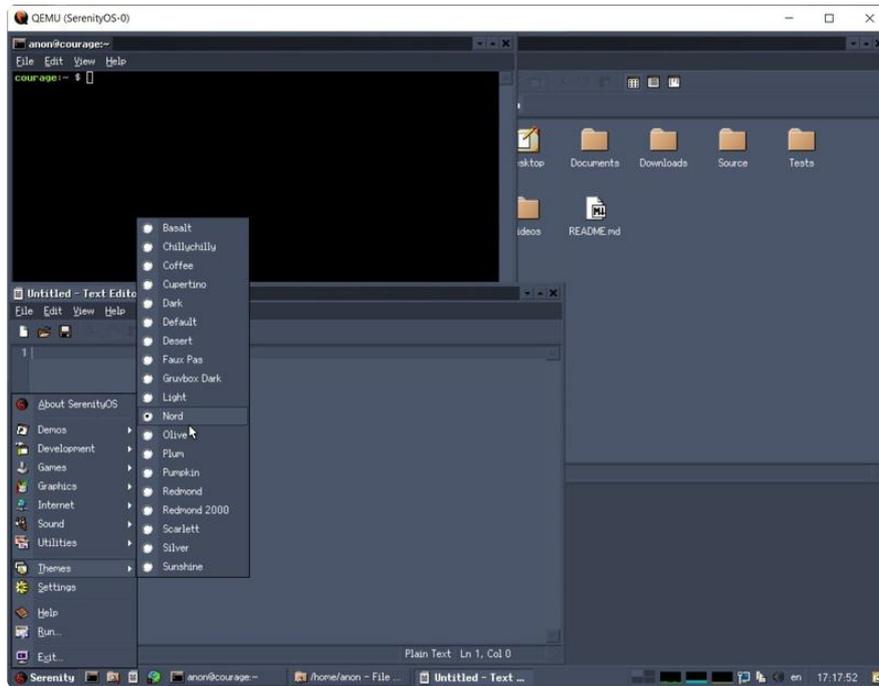
Running

We did it, we built the OS! Now what do we do?

Check it out! It's a full-featured system with a web browser, games, productivity apps, everything. Explore and see what you can find.







Keep in mind it's still in development, so things may not work perfectly. If you'd like to help make it better, you can! SerenityOS is open source and actively developed.

If you want to update to the latest code, you can do:

```
git pull
```

```
Meta/serenity.sh run
```

More Info / Contributing

You got it running and checked everything out, now what?

You could get things running in a [VirtualBox VM \(\)](#), or even try it on [real hardware \(\)](#).

Check out the GitHub page for more! [[https://github.com/SerenityOS/serenity \(\)](https://github.com/SerenityOS/serenity)]

SerenityOS is written in C++, so if you're familiar with it you can contribute code. Make sure you read [CONTRIBUTING.md \(\)](#) first and follow the policies there.

You don't have to be a programmer, you can also contribute art! [[https://github.com/SerenityOS/artwork \(\)](https://github.com/SerenityOS/artwork)]

Check out the [Discord \(\)](#) and chat with the community!

You can even watch the developers on YouTube: [Andreas \(\)](#), [Linus \(\)](#)

Or just enjoy it :)

