If you want the output from your executable to be displayed in a terminal window immediately after login, you can achieve this by creating a custom desktop entry that runs your program. Here's how you can do it:

1. \*\*Create a Desktop Entry\*\*:

Create a `.desktop` file that defines the program you want to run. You can use a text editor to create a file, e.g., `my\_executable.desktop`:

```ini

[Desktop Entry]

Type=Application

Terminal=true

Name=My Executable

Exec=/path/to/my\_executable

```

Replace `/path/to/my\_executable` with the actual path to your executable.

The `Terminal=true` line tells the system to open a terminal window when running the executable.

2. \*\*Move the Desktop Entry\*\*:

Move the `.desktop` file to the `~/.config/autostart/` directory:

```shell

mv my\_executable.desktop ~/.config/autostart/

```

3. \*\*Log Out and Log Back In\*\*:

After logging out and logging back in, your program should be automatically run in a terminal window.

Please ensure that your executable has the necessary permissions to run and that the `Terminal=true` line is present in your `.desktop` file to open a terminal window for your program.

To create a desktop entry that opens a terminal window and runs your executable immediately after login, follow these steps:

1. \*\*Create a Desktop Entry File\*\*:

Create a `.desktop` file in your home directory. You can name it, for example, `my\_executable.desktop`. Use a text editor to create the file:

```bash

nano ~/my\_executable.desktop

```

Add the following content to the file, making sure to replace `/path/to/your/executable` with the actual path to your executable:

```ini

[Desktop Entry]

Type=Application

Name=My Executable

Exec=gnome-terminal -- /path/to/your/executable

```

This `gnome-terminal --` command instructs the desktop entry to run your executable in a terminal window.

2. \*\*Make the Desktop Entry Executable\*\*:

To ensure the desktop entry file is executable, run:

```bash

chmod +x ~/my\_executable.desktop

```

3. \*\*Move the Desktop Entry File to the Autostart Directory\*\*:

Move the desktop entry file to the `~/.config/autostart/` directory:

```bash

mv ~/my\_executable.desktop ~/.config/autostart/

```

4. \*\*Log Out and Log Back In\*\*:

After you log out and log back in, the system should automatically open a terminal window and run your executable, displaying its output.

This approach uses the `gnome-terminal` command to open a terminal window and run your executable. Make sure to adjust the desktop entry file's contents and name according to your needs. If you're using a different desktop environment, you may need to specify the terminal emulator relevant to your desktop, such as `x-terminal-emulator` for Xfce or `konsole` for KDE, instead of `gnome-terminal`.

To achieve the goal of running `key\_check` and `key\_loader` services after login and displaying their output in terminal windows, you can create two systemd user services with dependencies. Here are the detailed steps:

1. \*\*Create a systemd service unit file for `key\_check`\*\*:

Create a service unit file for `key\_check` in your home directory (e.g., `key\_check.service`). Use a text editor to create the file:

```bash

nano ~/.config/systemd/user/key\_check.service

```

Add the following content to the file, and replace `/path/to/key\_check` with the actual path to your `key\_check` executable:

```ini

[Unit]

Description=Key Checker Service

[Service]

Type=simple

ExecStart=/path/to/key\_check

StandardOutput=tty

[Install]

WantedBy=default.target

```

2. \*\*Move the service unit file to the user systemd directory\*\*:

Move the `key\_check` service unit file to `~/.config/systemd/user/`:

```bash

mkdir -p ~/.config/systemd/user

mv key\_check.service ~/.config/systemd/user/

```

3. \*\*Create a systemd service unit file for `key\_loader`\*\*:

Create a service unit file for `key\_loader` in your home directory (e.g., `key\_loader.service`). Use a text editor to create the file:

```bash

nano ~/.config/systemd/user/key\_loader.service

```

Add the following content to the file, and replace `/path/to/key\_loader` with the actual path to your `key\_loader` executable:

```ini

[Unit]

Description=Key Loader Service

After=key\_check.service

[Service]

Type=simple

ExecStart=/path/to/key\_loader

StandardOutput=tty

[Install]

WantedBy=default.target

```

4. \*\*Move the service unit file to the user systemd directory\*\*:

Move the `key\_loader` service unit file to `~/.config/systemd/user/`:

```bash

mv key\_loader.service ~/.config/systemd/user/

```

5. \*\*Start and enable both services\*\*:

Start and enable both services as user services:

```bash

systemctl --user start key\_check.service

systemctl --user enable key\_check.service

systemctl --user start key\_loader.service

systemctl --user enable key\_loader.service

```

6. \*\*Log Out and Log Back In\*\*:

After logging out and logging back in, the `key\_check` service will run first and display its output in a terminal window. Once `key\_check` completes, the `key\_loader` service will run and display its output in a separate terminal window.

This setup ensures that you see the output of both services in terminal windows immediately after you log in. Make sure to replace the placeholder paths in the service unit files with the actual paths to your `key\_check` and `key\_loader` executables.