Written by Archer Heffern
For the System Creation Course

Database in C Deployment

Goal

We would like to deploy our database

Modify the existing database project with the following deployment features:

1. Make the database run as a service
2. Modifying the interface to allow for remote processes to interact with it

Make the Database a Service:

Make the database run as a service through your computer's init system.

To do this, create a service file for whatever init system your computer is using.(see (5)systemd.service if using systemd) The following features must be supported:

- Description
- runs on system startup
- restarts on failure
- runs after network service open

As reference, I would suggest looking at the other service files on your computer

Once the service file is created and moved to the proper location, ensure the database service starts automatically upon system boot or through a startup script. This ensures the database is always available as a background process.

Remote Process Interaction:

Modify the database interface to support remote process interaction. A simple networking library using sockets, along with some examples is provided.

See prerequisite information for how to test and other useful networking information.

This should be fairly simple, as you only need to change the input aspect of your program. interesting considerations are how much functionality should be provided? This is the question of where the line between frontend and backend is. Does the socket provide a full user experience? Or just respond to single function requests before closing the connection? In that case, we should probably create a client program responsible for connecting and providing a clean user interface.

Written by Archer Heffern
For the System Creation Course

Prerequisite Information

Services

A service is a process continuously running in the background.

Init Systems

The init system is crucial as the first process on a computer, responsible for initiating and managing all other processes.

Init Systems:
Linux:

- Default: systemd (managed via systemctl)
- Older versions: systemV (managed with the service command)

WSL (Windows Subsystem for Linux):

- Uses systemV for performance reasons.

Mac:

- Utilizes launchctl.

Init systems typically rely on configuration files to specify how services should be started and managed.

Computer Networking Review and Testing

IP Addresses
Every computer with internet access has a few IP addresses it recognizes and can listen to, including, but generally not limited to:
127.0.0.1 - Itself
0.0.0.0 - All IP addresses
<public/private IP> - address visible to other computers on the network, and possibly the entire world. Typically you have a private IP. View this using `ifconfig` or `ipconfig` (windows)

Run the dumbfetch example using google's IP address *142.251.32.110* (gotten via `dig google.com` - see ANSWER SECTION) and port *80*.

Testing
Use `netcat` to connect to the server and send sample messages
Check if a server is running using `ping`
Use `curl` to make requests using high level protocols and download server output.
Use `dig` to translate domain name to IP address (We have not covered domain names yet)

Written by Archer Heffern
For the System Creation Course

Submission

Create a git repo to track your project, push it to github, and send the git repository URL to me.

Below is the list of files expected in the git repository:

1. service file
2. Database source code
3. .gitignore : Should list all executables
4. start.sh or makefile : For compiling project
5. README.md : A readme file containing instructions for compiling, running, and any other information
6. Include any other helper files used in your project