Building the EmulationCore on a Linux-Debian System

EmulationCore is the "prototype" of the Voice Control System for IVI.

The EmulationCore was developed on a Windows PC, so the initial configuration was done to fit the Windows Environment. In order to setup the project inside Linux-Debian Environment, you may follow the instructions below.

Repository Folder: https://github.com/GENIVI/GENIVI-GSoC-18/tree/master/EmulationCore

Instructions for Setting up the Project

Requirements

1. **Python 3.6.5**: You should have a decent Python 3.6.5 installation on your debian system. If you do not have, please consult this Q&A in stackoverflow.
2. **Python-Dev Tools**
   a. Python-Dev Tools contain the header files you need to build Python extensions. They usually compile C/C++ code which are developed as python extensions. In our case, this is used to compile some packages written in C, used in PyAudio which outputs the audio.
   b. Run the following command to get them installed.

   ```bash
   sudo apt install python3.6-dev
   ```
3. **Python Virtual Environment Tools**
   a. We are supposed to get everything working inside a python virtual environment, so that our system is independent of external issues, specially with dependencies.
   b. Run the following command to install python venv tools for linux.

   ```bash
   sudo apt install python3-venv
   ```
4. **Pip3 Python Package Management Tools**
   a. All the dependencies are installed via pip3 inside our virtual environment so, package management tools are essential.
   b. Note that we need pip3 instead of pip as we are using Python 3.6.5
   c. Run the following command to install pip3

   ```bash
   sudo apt-get -y install python3-pip
   ```
5. **Driver Extensions, Audio Stream Handling libraries and other Developer Extensions**
   a. We would need certain libraries and extensions on the system side for TTS and STT to perform their job.
   b. Run the following command to get them installed.

   ```bash
   sudo apt install libasound-dev portaudio19-dev libportaudio2 libportaudio2-cpp0 ffmpeg libav-tools
   ```
6. **Dependencies of Processes**
   a. Similar to the above dependencies being related to EmulationCore, there may be some dependencies required to be installed by the Processes.
   b. Please follow the guide here: Prerequisites for Processes to get them installed.
   c. Return back here once you’ve completed installing them.
7. **Dependencies of Task Executors**
   a. Similar to the above dependencies being related to EmulationCore, there may be some dependencies required to be installed by the Task Executors.
   b. Please follow the guide here: Prerequisites for Task Executors to get them installed.
   c. Return back here once you’ve completed installing them.

Cloning the Project

- Navigate into the folder where you need to clone the project in a terminal.

  ```bash
  cd */path/to/folder/to/clone*
  ```
- To clone the repository, just run the following command.

  ```bash
  git clone https://github.com/GENIVI/GENIVI-GSoC-18.git
  ```
Migration/CMUSphinx: If you need to continue with CMUSphinx for Speech Recognition, follow the instructions in this page. Wait until everything’s completed.

Creating the Virtual Environment

- We must create our virtual environment inside the EmulationCore folder of the folder you’ve cloned the project.
- Navigate into the EmulationCore directory in the directory you’ve cloned the entire repository.

```
cd */path/to/folder/to/clone/EmulationCore*
```

or if you’re currently inside the cloned directory,

```
cd EmulationCore
```

- Initialize a virtual environment as follows.

```
python3 -m venv python_venv
```

- Activate the virtual environment

```
source python_venv/bin/activate
```

Installing the Dependencies

**Packages from PyPI**

- Some of the requirements are specified inside the requirements.txt in EmulationCore folder.
- Just run the following command in the activated virtual environment(in terminal) to get them installed.

```
pip3 install -r requirements.txt
```

- Note that every dependency will be installed only for the virtual environment, so they won't be globally available.

**Packages need to be Compiled**

- In order to install other requirements which might be need to be compiled under own environments, please follow the guides:
  1. Dependencies for Processes.
  2. Dependencies for Task Executors.
- Please note that this step is also vital for the EmulationCore to work as it would contain must-have set of dependencies for it work.

Installing NLTK Tokenizer

- We need to install a word tokenizer for NLTK library.
- NLTK has a set of tokenizers, we’re supposed to use "punkt" as our tokenizer.
- Run the following commands to get it installed.
- Note: This installation may take a while.
- First get a python console in your terminal(inside activated virtual environment).

```
python3
```

- Import nltk and download the punkt tokenizer

```
import nltk
nltk.download("punkt")
```

- Get rid of the python console by typing the following command, which returns you back to the bash.

```
exit()
```

Changing the Configuration
Please follow the instructions given [here](#) to change the configuration of EmulationCore.

*Note: This must be done, otherwise EmulationCore would cease to function properly.*

**Testing the Program**

- Just run the following command from the EmulationCore directory, in activated virtual environment to get the system running.

  ```
  python3 base/run.py
  ```

If you've completed every step correctly, program will run fine.

Please note that these instructions are only for Debian Linux system, and **NOT** for all linux systems. The functionality would not be guaranteed on other systems currently.

Good Luck! 🍀