What is `jobinfo`?

Researchers on a Linux-based computing system submit computing jobs to run and process their research data. Information on these jobs is collated in an output table as they run and provide a record of the resources used by the job. `jobinfo` extracts useful information from such tables, including,

- Allocation code(s) corresponding to a user.
- Most recently submitter job ID.
- Total CPU usage in core-hours.
- Total memory usage in GB.

Installation Instructions

Prerequisites

- Ensure Python 3.6+ is installed.

```
python --version
```

or

```
python3 --version
```

- If you don't have Python 3.6+ installed, go to the official webpage and follow the instructions to install the latest version.
- `pip` should come with Python, but ensure it's up to date.

```
python -m pip install --upgrade pip
```

Option A: `pipx` Installation (recommended)

- Install the latest version of `pipx`.
- Run the following commands to install `jobinfo`.

```
pipx install jobinfo
pipx upgrade jobinfo
```

Option B: Virtual Environment Installation (recommended)

- Create and activate a virtual environment.

```
python -m venv testenv
source testenv/bin/activate  # Linux/Mac
testenv\Scripts\activate     # Windows
```

- Run the following command to install `jobinfo`.

```
pip install --upgrade jobinfo
```

- After your work is done deactivate the virtual environment.

```
deactivate
```

Option C: Global Installation (not recommended)

- Use `pip` package manager.

```
pip install --upgrade jobinfo
```

- For a user-specific install (to avoid system-wide install),

```
pip install --user jobinfo
```

Option D: Build From Source

- Download the source code either using the following command or as a `.zip` package.

```
git clone https://github.com/ahama92/jobinfo.git
cd jobinfo
```

- You may choose to use a virtual environment (recommended) or install globally (not recommended) just like before.

```
pipx install .
```

or

```
pip install .
```

Installation Confirmation

- Check if `jobinfo` is installed correctly.

```
jobinfo --version
```

Troubleshooting

If you use Windows, I highly recommend installing the Ubuntu terminal environment from Microsoft Store. This app provides a lot of what a Linux terminal has to offer. You can do all of your research work and computations from this terminal. If for some reason, you still want to use Windows and you face issues with running `jobinfo`, here are some possible remedies.

- Check if `jobinfo` was installed.

```
pip show jobinfo
```

- The output should show the installation location similar to `c:\users\USER\appdata\local\packages\pyt`
- Go to that path in your file explorer.
- Then go one step up. In my example it would be the `python39\` folder.
- Then go to the `Scripts` folder.
- Copy the path.
- Then type the following command in a Windows PowerShell.

```
$env:Path += ";C:\Users\USER\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5
```

- Don't forget to change the path to that one you just copied, not my example!

Software Prerequisites

- Python 3.6 or higher.

Usage Guide

Basic Usage

```
jobinfo FILENAME -u USER [-a] [-r] [-c] [-m] [-s] [-v] [-h]
```

Positional Arguments

- `FILENAME` job information file in CSV format.

Options

- `-u USER`, `--user USER` the username.
- `-a`, `--alloc` show allocation code(s).
- `-r`, `--recent` show most recent job ID.
- `-c`, `--cpu` show total CPU usage in core-hours.
- `-m`, `--mem` show total memory usage in GB.
- `-s`, `--simple` simple output formatting (no table).
- `-v`, `--version` prints version information and exits.
- `-h`, `--help` shows the help message and exits.

Examples

Let's try out a few examples with a `test.csv`. This files contains the following information.

Username	Allocation	JobID	CPUs	JobDuration	Memory
user1	alloc-1	111111	16	1800	180
user2	alloc-2	222222	8	600	36
user2	alloc-3	333333	32	480	600

- **Test-1:** Extract the allocation code(s) for user `user1`.

```
jobinfo test.csv -u user1 -a
```

- **Test-2:** Extract the total CPU and memory usage for user `user2`.

```
jobinfo test.csv -u user2 -cm
```

Figure 1: Test-1

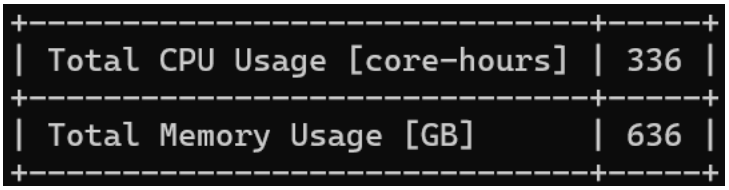

Figure 2: Test-2

- **Test-3:** Extract allocation code(s), most recent job ID, as well as total CPU and memory usage for user `user2`. Print the output in simple formatting with no table.

```
jobinfo test.csv -u user2 -arcms
```

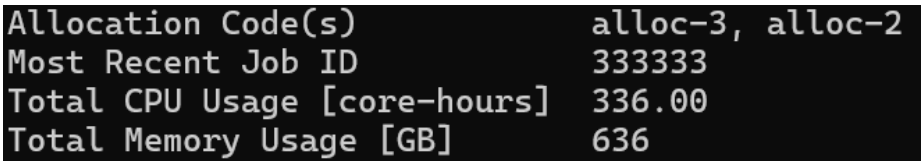

Figure 3: Test-3

Notes

Make sure the input file is in CSV format with exactly the following header,

```
Username,Allocation,JobID,CPUs,JobDuration,Memory
```