Day 13-1

Assignment:

- HW 06 – Due today
- Projects – Weekly report due 8am Thursday

Today's Topics:

- install.sh and setup.sh
- tmux
- Log files
- autostart

13-1 Before You Ship

What to do when you think you are done

Once things are working...

- **`install.sh`** and **`steup.sh`**
- Long term testing
- Log files
- autostart

Install and setup

- Create an install.sh: **`touch install.sh`**
  - Make it executable: **`chmod +x install.sh`**
  - Put in it instructions for installing additional software needed by your project. Things that are only done once
  - **`apt-get install this and that`**
  - **`npm –g install bone`**
  - **`wget http://my.software.com/`**
- Create **`stepup.sh`** and make it executable
  - Put in it instructions for running your project. Things that are only done every time you reboot the Bone, etc.
  - **`export MYPATH=/root/here`**
  - **`cp project.js /var/run/cloud9/autorun`**

Long Term Testing

- What happens when you:
  - Connect via **ssh**
  - Start your project: **`start.sh &`**
  - Then quit ssh?
- Your project is killed

Long Term Testing

- Use **`tmux`**
- Demo…

- Google tmux tutorial
- I like: https://danielmiessler.com/study/tmux/

Once things are working...

- `install.sh` and `steup.sh`
- Long term testing
- Log files
- autostart

Log files - Winston

- You can use **`console.log`** to record your project's activities
- But there are better ways...
- Try Winston: https://www.npmjs.com/package/winston
- **`npm install winston`**

```
var winston = require('winston');

winston.log('info', 'Hello distributed log files!');
winston.info('Hello again distributed logs');
```

- Displayed on console

Logging levels and output

- `{ error: 0, warn: 1, info: 2, verbose: 3, debug: 4, silly: 5 }`
- `winston.debug('Hello again distributed logs');`

- You can direct the logging to various places
  - `winston.add(winston.transports.File, { filename: 'somefile.log' });`
- Or turn it off
  - `winston.remove(winston.transports.Console);`

- See https://www.npmjs.com/package/winston for details
- Or google `winston tutorial node`

Autostart

- Most project should autostart once the Bone is powered up
- Three ways to autostart:
  - systemd
  - /var/lib/cloud/autorun
  - crontab

systemd

- Covered earlier
- See: http://elinux.org/EBC_systemd
- Or google **`systemd tutorial`**

/var/lib/cloud9/autorun

- Any files placed in **/var/lib/cloud9/autorun** will automatically run
- If the file is edited, the program will be restarted
- Files ending with **.js**, **.py** or **.ino** all work

/etc/crontab

- The file **/etc/crontab** is a way to schedule things to be run

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
# My Stuff
NODE_PATH=/usr/local/lib/node_modules
* * * * * root /root/exercises/phant/weather.js

* * * * * root echo "Test message" 2>&1 |logger
```

Once things are working...

- **`install.sh`** and **`steup.sh`**
- Long term testing - tmux
- Log files - winston
- autostart

- Assign someone on your team to look into each one of these
- Report who is doing what in your weekly memo