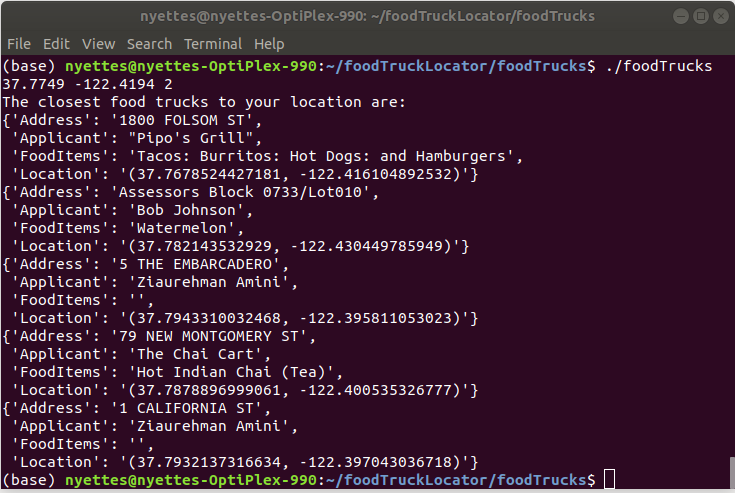
## Summary/Notes on Food Truck Locator Exercise

* First thought was to get a distance between 2 points using latitude and longitude. But then needed to adjust that line of thinking because the second point is unknown.
* Next is to load the data into a database so it can be queried based on locations (latitude/longitude) that fall within a certain radius.
  + Decided to use MongoDB since it uses JSON documents and I have been working on learning more about NoSQL databases.
    - Reviewed CSV file to determine what fields are available and what datatypes should be used.
    - Lost some time trying to do in Linux (the interface is not as fleshed out as the Windows one). Could have used command line, but it was going to take longer as I am still getting familiar with the platform.
    - Went back to Windows because the API was going to be built using Visual Studio (still more comfortable using it even though it has been a while since I used it to build an interface)
    - Went back to Linux as not all functions that I wanted to use were available. Linux ended up being faster, but not portable to Windows
  + From originating location, locate 5 closest trucks starting with 1 mile until at least 5 trucks are found (did not actually implement this)
    - Need to convert lat/long to radians
    - Check if originating location is within the city
    - Calculate max/min points
    - Convert back to degrees
    - Query
      * Fields to return, Sort and Count
      * Adding Calculated fields (Latitude and Longitude in Radians) and radius from origination point
* From results, output to CLI or Web interface using JSON (only implemented JSON output)
* Tests (Only one run the city center)
  + Use Official (center of the city) San Francisco Latitude and Longitude 37.7749° N, 122.4194° W image of results posted on README
  + Use another set of coordinates in the city Pier 39 37.8087° N, 122.4098° W
  + Use coordinates of one of the food trucks
  + Use out of Range (Seattle) 47.6062° N, 122.3321° W
* Further development/lessons learned
  + Take maximum connections and timeout parameters into consideration when connecting to the database. Also, security considerations and configuration files
  + Run on Windows
  + Probably should not use a database platform that I am not as comfortable with for a task such as this where there are time constraints. Creating the radius from the originating point expression in the pipeline was challenging, but I probably should have scrubbed the data first.



Output for 37.7749° N, 122.4194° W