Design and Implement Loadable Kernel Module and User Space Module

Team members: Prachi Shirode (819718205)
Raksha Gopal Kulkarni (820935343)

Project plan: We are planning to implement a loadable kernel module and a user space module for a task and compare their performance based upon the execution time. The user space module is designed for incubator which includes reading the data from sensor and averaging the read values and displaying the state of incubator with LEDs. The same will be implemented in the loadable kernel module.

Application areas: The temperature and humidity sensors play a very important role in various platforms like incubators, air cooling systems, greenhouse, weather department, aircrafts, spacecraft, and scientific labs. The temperature and humidity application in an aircraft or incubator forms a critical parameter for the system. Hence, measuring them accurately and adjusting the variations becomes very important and prominent.

Accomplishments:

Significance of selected hardware components: Raspberry Pi 3, temperature and humidity sensor (DHT22) (3 in number), breadboard, 10kΩ resistor, 220Ω resistor, LEDs (3 in number) and jumper wires.

- **Raspberry Pi 3:** This is the third generation model of raspberry pi. Compared to other generations, this model is 10x times faster. We chose raspberry pi for our project because it is widely used for project like reading values from sensors which is easy to understand with lot of information available online.

- **DHT22 sensor:** We had option of selecting sensor DHT11 or DHT22. We found that DHT22 is more accurate compared to DHT11. The temperature and humidity range is high and more accurate in DHT22. The size of this sensor is small with low consumption and long transmission distance (20m) which makes DHT22 suitable to perform its best even in harsh case applications. DHT22 has exclusive digital signal collecting technique and humidity sensing technology which contributes to its reliability and stability. DHT22 has 4 pins in a single row. From left to right, the pin 1 is the VDD power supply (3.3 – 6 V DC), pin 2 is data signal, pin 3 is null and pin 4 is grounded. The temperature and humidity values are measured once every 2 seconds.

Software implementation:

- Operating system used on Pi: Raspbian Jessi
- Tools used:VNC, Git, WinSCP, GCC compiler
- Programming language used: Python, C

Experiments performed:

User space: We implemented the program for incubator to read the temperature and humidity values from the sensor DHT22 using python. The python program reads temperature and humidity values from the three sensors which are connected in parallel. The average value of temperature and humidity is computed.

Depending upon the average value, the LEDs are lit. Generally, the temperature should be in between 37 C to 40 C and humidity should be in between 50 to 55 for an incubator which is shown by the green LED. If the temperature and humidity value goes above the desired value, the red LED is lit signaling for human intervention. Similarly, if the value goes below the desired value, the yellow LED is lit. Also, the time required to run the program is also displayed.

Kernel space: We implemented the program of 'hello world' and 'goodbye world' in the kernel space. The insmod and rmmod commands were used to insert and remove the module in the kernel respectively. To see the output of the program, the dmesg command needs to be used. The dmesg is used to print the message buffer of the kernel.

Results:

User space output:

```
Reading from 1st sensor: Temp=25.8*  Humidity=44.8%
Reading from 2nd sensor: Temp=25.8*  Humidity=43.7%
Reading from 3rd sensor: Temp=25.8*  Humidity=46.4%
Average temperature is: Temp=25.8*  Humidity=45.0%
led.py:36: RuntimeWarning: This channel is already in use, continuing anyway.  U
se GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(21, GPIO.OUT) ## Setup GPIO Pin 7 to OUT
Time taken is: --- 5.36941599846 seconds ---
```

Kernel space output:

```
[ 3017.680665] hellokernel: loading out-of-tree module taints kernel.
[ 3017.680681] hellokernel: module license 'unspecified' taints kernel.
[ 3017.680686] Disabling lock debugging due to kernel taint
[ 3017.681832] Hello world 1.
```

```
[ 3017.680665] hellokernel: loading out-of-tree module taints kernel.
[ 3017.680681] hellokernel: module license 'unspecified' taints kernel.
[ 3017.680686] Disabling lock debugging due to kernel taint
[ 3017.681832] Hello world 1.
[ 3279.623168] Goodbye world 1.
pi@raspberrypi:~/src/c/lkm $
```

Milestones covered:

Proposed Deliverable	Proposed Date	Delivered Date	Comments
1. Remotely accessed Linux server using VNC. 2. Drafting project proposal.	Sept 17th, sun	Sept 17th, sun	Completed
3. Final editing of project proposal and procuring the sensors and studying its datasheet and acquiring knowledge to for creating user space module for sensor. 4. Interfacing sensor and developing user module.	Sept 24th, sun	Sept 24th, sun	Completed
5. Development continued	Oct 1st, sun	Oct 1st, sun	Completed
6. Running test iterations and observing values	Oct 8th , sun	Oct 8th , sun	Completed
7. Buffer period	Oct 15th ,sun		N/A
8. Midterm week	Oct 22nd,sun		N/A
9. Drafting project progress demo and preparation of presentation 10. Project progress demo	Oct 29th, sun Oct 30th, mon	Oct 29th, sun Oct 29th, mon	Completed
11. Understanding the process of making modifications to raspbian kernel using LKM.(hello world)	Nov 5th, sun	Oct 27th, fri	Completed

Future milestones:

Week	Date	Task
Week 11	Nov 12th,sun	1. Implementing LKM
Week 12	Nov 19th,sun	2. Implementing LKM continued (buffer period)
Week 13	Nov 26th,sun	3. Running test iterations and observing results.
Week 14	Dec 3rd,sun	4. Comparing test results of both the modules
Week 15	Dec 10th,sun	5. Drafting Final Project Report and proof reading
Week 16	Dec 11th, mon	6. Final project demo

References:

https://www.lifewire.com/light-an-led-with-the-raspberry-pis-gpio-4090226

http://www.instructables.com/id/Raspberry-PI-and-DHT22-temperature-and-humidity-lo/

https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf

http://www.circuitbasics.com/how-to-write-and-run-a-c-program-on-the-raspberry-pi/

https://raspberrypi.stackexchange.com/questions/39845/how-compile-a-loadable-kernel-module-without-recompiling-kernel