

Temperature Monitoring with the Linux Kernel on a Multi Core Processor

R.Arvind¹, Mrs.A.Vinnarasi²P.G. Student, Department of Electronics and Communication Engineering, SRM University, SRM Nagar,
Kattankulathur, Chennai, Tamil Nadu, India¹Assistant Professor, Department of Electronics and Communication Engineering, SRM University, SRM Nagar,
Kattankulathur, Chennai, Tamil Nadu, India²

ABSTRACT:This paper presents a method of collecting temperature information timely and accurately from the thermal sensors in a quad core processor by patching the Linux Kernel with the CoreTemp module. Modern multi core processors are prone to overheating, which can be reduced and prevented by appropriate temperature monitoring. We also present the technique used to read the temperature into a file by using file operation commands for purposes such as temperature aware scheduling.

KEYWORDS:Temperature, Linux, Multicore, Scheduling, Thermal-Aware.

I. INTRODUCTION

With the recent advancements in nanometre semiconductor technology, more powerful processors are being fabricated to be used in devices such as smartphones, tablets, laptops and desktop computers. As a result more heat is generated which increases the thermal stress on the internal components of these devices. The resulting heat is transferred to the outside world through the external casing, causing discomfort to the user holding it in the case of a smartphone or tablet. In laptops and desktop computers, heat is dissipated through fans which are quite noisy. Also with an increasing number of cores, comes increased power consumption which in turn results in elevated temperatures since power and temperature share a linear relationship. Choi et al. [4] observed that a few degrees increase in temperature brings down the lifetime by 50% for electronic circuits. High temperature affects system reliability resulting in Electromigration, stress migration and dielectric breakdown.

Processor temperature should be carefully maintained within a proper working temperature limit to avoid hardware damage and performance degradation. A thermal sensor integrated on the microprocessor is the major component to obtain the system temperature information. It is used to guide the operating system to manage the cooling system. For example, the operating system reads the current temperature value from the thermal sensor, and based on the temperature, a new fan speed is calculated for the system configuration. An alternative method is to read temperature value directly from the built-in digital thermal sensor integrated with each core, which can provide more accurate temperature information of the core. The temperature value is stored in the Model Specific Register (MSR), which can be accessed through the Industry Standard Architecture (ISA) ports or the System Management Bus (SMB). To ease our implementation and tests, we simply adopted a Linux hardware monitoring tool called lm-sensors [2] to capture the temperature.

Paper is organized as follows. Section II presents the related work that has been done. Section III presents a brief description of the hardware and software used and goes on to explain in detail the Digital Thermal Sensor. Section IV describes the method of installing CoreTemp module. Section V deals with the methods to obtain temperature to use in a program. Finally, Section VI presents conclusion.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2015

II. RELATED WORK

Heat sinks and cooling fans are commonly used technology for cooling IC chips in computer systems. They are designed based on the concept of controlling the airflow rate between the heat sink and the ambient, so that the computing system can operate under the desired temperature. The heat sinks are made of metal with high thermal conductivity, and are directly connected with the small processor chip. They allow heat to transfer to its large surface area by conduction. Then, the computer fans remove the heat from the heat sink to the ambient environment. The thermal performance of the heat sink depends on many parameters, such as the thickness, surface area, and thermal conductivity. Once the heat sink is manufactured, its cooling efficiency is fixed. Thus, the air-cooling performance is adjusted by changing the fan speed.

Usually in today's desktop computers, the fan speed is controlled by a closed loop controller, which regularly collects the temperature information from CPU thermal sensors, and adjusts fan speed accordingly to accommodate the corresponding temperature change and the power cost for cooling down computing systems is high. Moreover, the increasing fan speed can introduce system noise, which not only leads to an uncomfortable working environment, but also causes vibrations and impacts the reliability. Hence with temperature monitoring software techniques such as temperature aware scheduling were developed which overcame the limitations of mechanical cooling technologies. One primary approach adopted in operating system level for thermal management is thermal aware scheduling, which takes processor temperature into account to avoid thermal violation, and keeps the temperature below a certain limitation by throttling the "hot" tasks.

III. SYSTEM DESCRIPTION

A laptop running on Core i7 2630QM processor which has 4 cores and 8 threads is used as the experimentation platform. The Operating system chosen is Ubuntu 14.04 LTS (Long Term Stable) running Linux kernel version 3.18.1. Linux was chosen as the software platform since a lot of embedded devices, workstations and consumer products such as smartphones running Google's Android operating system use the Linux kernel.

The Digital Thermal Sensor

It is a serially accessible sensor particularly suited for low cost and small form factor applications. Temperature data is converted from the on-board thermal sensing element and made available as an 8-bit digital word. Communication is accomplished via a 2-wire SMBus/I2C compatible serial port. Temperature resolution is 1°C. Conversion rate is a nominal 8 samples/sec. Small size, low installed cost and ease of use make it an ideal choice for implementing thermal management in a variety of systems. [5]

The DTS acquires and converts temperature information from its on-board solid-state sensor with a resolution of $\pm 1^\circ\text{C}$. It stores the data in an internal register which is then read through the serial port. The system interface is a slave SMBus/I2C port, through which temperature data can be read at any time. Eight SMBus/I2C addresses are programmable, which allows for a multi-sensor configuration. Also, there is low power Standby mode when temperature acquisition is suspended.

Register Set and Programmer's Model

D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
MSB	X	X	X	X	X	X	LSB

Table.1. Temperature Register

Table.1 Shows the format of the temperature register which consists of an 8 bit word starting from D[7] to D[0]. The binary value (2's complement format) in the temperature register represents temperature of the on-board sensor following a conversion cycle. The registers are automatically updated in an alternating manner.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2015

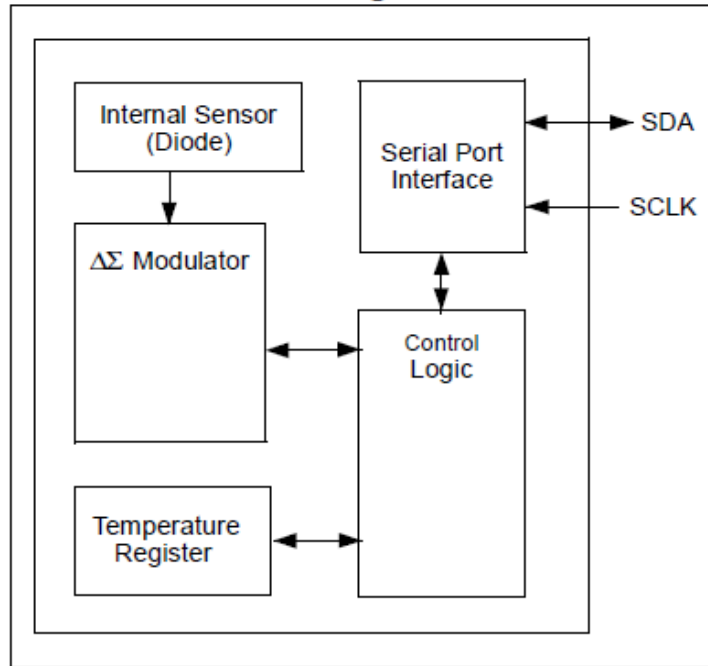


Fig.1. DTS Functional Block Diagram [5]

Fig.1 shows the schematic block diagram of a Digital Thermal Sensor. It consists of a diode which acts as an internal temperature sensor. A Modulator takes signals from the diode and converts analog temperature data into an 8 bit digital word which is stored in the temperature register. Conversion is accomplished with the help of a control logic which is driven by a serial port interface connected to Serial Data Access (SDA) and Serial Clock (SCLK) ports.

Real Temperature	Read Temperature	Binary Representation
130°C	127°C	01111111
127 °C	127°C	01111111
126.5°C	126 °C	01111110
25.25 °C	25 °C	00011001
0.5 °C	0 °C	00000000
0.25 °C	0 °C	00000000
0 °C	0 °C	00000000
-0.25 °C	-1 °C	11111111
-0.5 °C	-1 °C	11111111
-0.75 °C	-1 °C	11111111
-1 °C	-1 °C	11111111

Table.2. Temperature to Digital Value Conversion

Table.2 gives a detailed account of the temperature value conversion from analog to binary digital values. In temperature data registers, each unit value represents one degree (Celsius). The value is in 2's complement binary format such that a reading of 0000 0000b corresponds to 0°C.

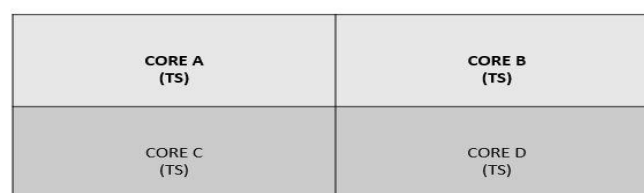
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2015

IV. PATCHING THE LINUX KERNEL WITH CORETEMP MODULE

A Core Temp patch [1] is added to the linux kernel to get the temperature of the cores through digital thermal sensors (DTS) located in each individual processing core near the hottest part as shown in Fig.2. The temperature readings are very accurate as this sensor is digital meaning it doesn't rely on an external circuit located on the motherboard to report temperature. Its value is stored in a Model Specific Register (MSR) in the processor so that software can access and read it. This eliminates any inaccuracies that can be introduced by external motherboard circuits and sensors.



TS- Temperature Sensor

Fig.2. Location of DTS in Cores

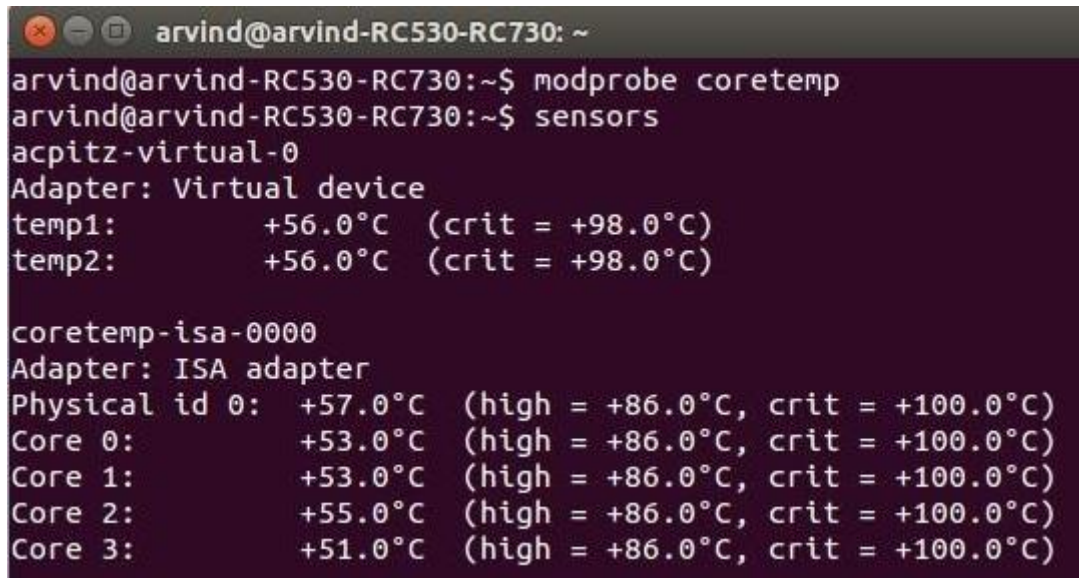
Outlined below are steps that explain the method of compiling the Coretemp patches [1] to get the Core i7 CPU Digital temperature sensors to work with Ubuntu version 14.04 LTS.

- Install the required development packages by typing the below command into a linux terminal
- `sudo apt-get install build-essential gcc linux-headers-`uname -r``
- Make a directory to store the coretemp file and change into that directory.
- After this download the modified coretemp file and extract the patch from the coretemp.c file.
- Finally apply the patch to the kernel source.
- Next create a makefile and compile the kernel module
- Compile the kernel image.
- It is advised to back up all the data before installing the patched module because inserting a custom module can crash the current kernel and lead to loss of information.
- Install and test the module.

After patching the kernel with coretemp the Linux hardware monitoring tool `lm_sensor` [2] is used. It is open-source and provides drivers for monitoring temperatures, voltage, etc. `lm-sensors` uses device files exported by the kernel for CPU temperature. Reading what these device files have as backing variables in the kernel yields the temperature information.

- The command `sudo apt-get install lm-sensors` installs `lm-sensors` package.
- To detect hardware monitoring chips, type the following command as the root user: `sudo sensors-detect`
- An interactive program then walks through the process of scanning the system for various hardware monitoring chips and sensors, supported by `lm_sensors`. The program will help one determine which kernel modules need to be loaded in order to use `lm_sensors`. Accept the default values for all questions.
- Load drivers using `modprobe` command : `modprobe coretemp`
- To read the temperature from the Digital Thermal Sensor on the CPU type the `sensors` command in the terminal window.

After the above process we get the output as shown in Fig.3.



```
arvind@arvind-RC530-RC730: ~
arvind@arvind-RC530-RC730:~$ modprobe coretemp
arvind@arvind-RC530-RC730:~$ sensors
acpitz-virtual-0
Adapter: Virtual device
temp1:          +56.0°C  (crit = +98.0°C)
temp2:          +56.0°C  (crit = +98.0°C)

coretemp-isa-0000
Adapter: ISA adapter
Physical id 0:  +57.0°C  (high = +86.0°C, crit = +100.0°C)
Core 0:         +53.0°C  (high = +86.0°C, crit = +100.0°C)
Core 1:         +53.0°C  (high = +86.0°C, crit = +100.0°C)
Core 2:         +55.0°C  (high = +86.0°C, crit = +100.0°C)
Core 3:         +51.0°C  (high = +86.0°C, crit = +100.0°C)
```

Fig.3. Result of Patching the Linux Kernel with CoreTemp

As shown in Fig.3 individual temperatures of cores are displayed along with the maximum safe temperature allowed which happens to be 86°C in this case and a critical temperature beyond which the processor will suffer thermal breakdown which is 100°C for the Core i7 CPU. This critical temperature is also called the junction temperature T_{JUNCTION} which is the maximum possible temperature allowed at the processor die.

V. READING TEMPERATURE DATA FOR SCHEDULING

Liang et al. in [3] only give a brief description about coretemp patches. No details are provided in [3] as to how to get the temperature information in a program to perform scheduling. In this section we explore how to get the temperature readings in a program viz two approaches:

- 1) Using file operation commands such as `fopen()` and `fread()`
- 2) By the use of shell scripting

File Operation Commands

A program is written in C programming language using file operation commands [6] such as `fopen()` and `fread()`. The usage of the commands and the corresponding algorithm are described in detail as follows.

`fopen()`

*FILE *fopen(const char *path, const char *mode);*

The `fopen()` function is used to open a file. This function takes two arguments. The first argument is a pointer to a string containing name of the file to be opened while the second argument is the mode in which the file is to be opened. The modes and their corresponding description are given below.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2015

Mode	Description
r	Open text file for reading
r+	Open for reading and writing
w	Create text file for writing
w+	Open for reading and writing
a	Open for appending (writing at end of file)
a+	Open for reading and appending

The `fopen()` function returns a FILE stream pointer on success while it returns NULL in case of a failure.

fread()

The `fread()` function is used to read data from the given stream into the array pointed by `ptr`.

`size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream)`

The various arguments used in `fread()` function and their explanation are given as follows.

`ptr` : Pointer to a block of memory with a minimum size of `size*nmemb` bytes.

`size` : Size in bytes of each element to be read.

`nmemb` : Number of elements, each one with a size of `size` bytes.

`stream` -- Pointer to a FILE object that specifies an input stream.

The total number of elements successfully read is returned as a `size_t` object, which is an integral data type. If this number differs from the `nmemb` parameter, either an error occurred or the End Of File was reached. The core temperatures were found to be located in `/sys/devices/platform/coretemp.0/tempn_input` where `n` is the core number. For example to get the temperature of core number 1 open `/sys/devices/platform/coretemp.0/temp1_input`.

Algorithm for File Operation Program

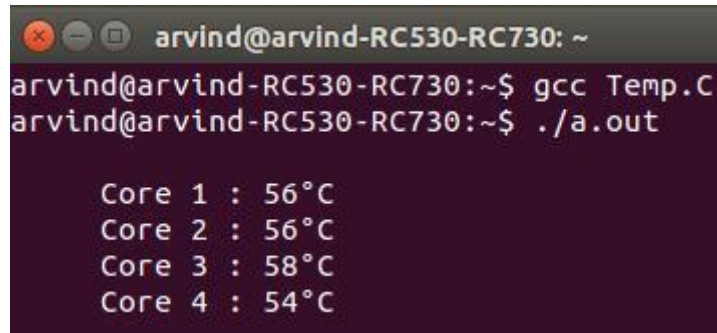
- Include necessary header files
- Define `size` as 1
- Define number of elements as 2
- Create file stream pointers for each core and initialize them to null and allocate memory
- Open the files containing the core temperatures in read mode and assign the values to corresponding file stream pointers for each core
- Read Core temperatures from opened files
- Print Core temperatures

Upon executing a program written using the above algorithm we get the output shown in Fig.4.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2015



```
arvind@arvind-RC530-RC730: ~  
arvind@arvind-RC530-RC730:~$ gcc Temp.C  
arvind@arvind-RC530-RC730:~$ ./a.out  
  
Core 1 : 56°C  
Core 2 : 56°C  
Core 3 : 58°C  
Core 4 : 54°C
```

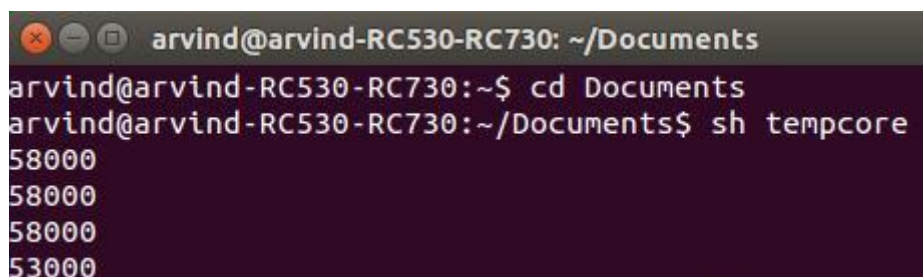
Fig.4. Output of Temp.C (a program to read temperature data from a file)

Shell Scripting

Another alternative to read temperature into a program is to use shell scripts [7]. Given below is a simple shell script to read and display the temperature of a core.

```
#!/bin/sh  
# temperature display of a core  
ADIR=/sys/devices/platform/coretemp.0/temp2_input  
ADIR1=/sys/devices/platform/coretemp.0/temp3_input  
ADIR2=/sys/devices/platform/coretemp.0/temp4_input  
ADIR3=/sys/devices/platform/coretemp.0/temp5_input  
  
cat $ADIR  
cat $ADIR1  
cat $ADIR2  
cat $ADIR3
```

Execute the above shell script using the command `sh tempcore`. It yields an output as shown in Fig.5.



```
arvind@arvind-RC530-RC730: ~/Documents  
arvind@arvind-RC530-RC730:~$ cd Documents  
arvind@arvind-RC530-RC730:~/Documents$ sh tempcore  
58000  
58000  
58000  
53000
```

Fig.5 Output of tempcore-a shell script to read temperature of cores

The values 58000 in the first three lines of the output are the temperatures of Core 1, 2 and 3 respectively while the value 53000 is the temperature of Core 4 as shown in Fig.5 above. The unit of temperature here is /1000°C.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2015

VI. CONCLUSION

First a research was conducted to study the side effects of adverse temperatures in embedded and handheld devices as well as mainstream desktop computers. Then a hardware study was made and the various functional components of the Intel Digital Thermal Sensor used to read temperature were detailed. Then the methods to patch the Linux Kernel with CoreTemp module to enable temperature reading was presented. Finally an algorithm to read the temperature of the 4 cores into a file was described in detail with the corresponding results of the operation shown via appropriate screenshots. This temperature monitoring is particularly useful in applications such as thermal aware scheduling which the authors are currently working on.

REFERENCES

- [1] <https://wiki.ubuntu.com/CoreTemp>
- [2] https://wiki.archlinux.org/index.php/lm_sensors
- [3] Liang Xia, Yongxin Zhu, Jun Yang, Jingwei Ye and ZonghuaGu “Implementing a Thermal-Aware Scheduler in Linux Kernel on a Multi-Core Processor”, Oxford University Press, 2010
- [4] J. Choi, C-Y. Cher, H. Franke, H. Hamann, A. Weger, and P. Bose “Thermal-aware task scheduling at the system software level” International Symposium on Low Power Electronics and Design, pages 213–218, 2007.
- [5] [Microchip TC74 Tiny Serial Digital Thermal Sensor Datasheet](#)
- [6] http://en.wikipedia.org/wiki/C_file_input/output
- [7] http://en.wikipedia.org/wiki/Shell_script