

hw0B. CPU temperature

1. 使用 strace 在你的處理器上追蹤 sensors 這個指令

```
s406410114 > (ssh) lonux ~ /OS_HW/hw0B.CPU_temperature strace sensors
execve("/usr/bin/sensors", ["sensors"], 0x7fff7d0a3ab0 /* 35 vars */) = 0
brk(NULL)                               = 0x55b9dcf35000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff0507c360) = -1 EINVAL (不適用的引數)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (沒有此一檔案或目錄)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=70372, ...}) = 0
mmap(NULL, 70372, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f71f309e000
close(3)                                = 0
```

2. 列出 sensors 從哪邊讀取 CPU 的溫度(使用 strace trace sensors 如何讀取 CPU 溫度擷圖)

(1)不斷進入此目錄去讀取 (/sys/devices/platform/coretemp.0/hwmon/hwmon4)

```
lstat("/sys/devices", {st_mode=S_IFDIR|0755, st_size=0, ...}) = 0
lstat("/sys/devices/platform", {st_mode=S_IFDIR|0755, st_size=0, ...}) = 0
lstat("/sys/devices/platform/coretemp.0", {st_mode=S_IFDIR|0755, st_size=0, ...}) = 0
lstat("/sys/devices/platform/coretemp.0/hwmon", {st_mode=S_IFDIR|0755, st_size=0, ...}) = 0
lstat("/sys/devices/platform/coretemp.0/hwmon/hwmon4", {st_mode=S_IFDIR|0755, st_size=0, ...}) = 0
lstat("/sys/devices/platform/coretemp.0/hwmon/hwmon4/device", {st_mode=S_IFLNK|0777, st_size=0, ...}) = 0
readlink("/sys/devices/platform/coretemp.0/hwmon/hwmon4/device", "../../../coretemp.0", 4095) = 19
```

(2) cd /sys/devices/platform/coretemp.0/hwmon/hwmon4 ls 出檔案

```
s406410114 > (ssh) lonux /sys/devices/platform/coretemp.0/hwmon/hwmon4 ls
device      temp1_crit_alarm  temp2_crit_alarm  temp3_crit_alarm  temp4_crit_alarm  temp5_crit_alarm
name        temp1_input       temp2_input       temp3_input       temp4_input       temp5_input
power       temp1_label       temp2_label       temp3_label       temp4_label       temp5_label
subsystem   temp1_max         temp2_max         temp3_max         temp4_max         temp5_max
temp1_crit  temp2_crit       temp3_crit       temp4_crit       temp5_crit       uevent
```

(3)cat name 出現 coretemp

```
s406410114 > (ssh) lonux /sys/devices/platform/coretemp.0/hwmon/hwmon4 cat name
coretemp
```

(4)cat temp_label 可以發現各個 CPU 被標記了不同代號

```
s406410114 > (ssh) lonux /sys/devices/platform/coretemp.0/hwmon/hwmon4 cat name
coretemp
s406410114 > (ssh) lonux /sys/devices/platform/coretemp.0/hwmon/hwmon4 cat temp1_label
Package id 0
s406410114 > (ssh) lonux /sys/devices/platform/coretemp.0/hwmon/hwmon4 cat temp2_label
Core 0
s406410114 > (ssh) lonux /sys/devices/platform/coretemp.0/hwmon/hwmon4 cat temp3_label
Core 1
s406410114 > (ssh) lonux /sys/devices/platform/coretemp.0/hwmon/hwmon4 cat temp4_label
Core 2
s406410114 > (ssh) lonux /sys/devices/platform/coretemp.0/hwmon/hwmon4 cat temp5_label
Core 3
```

3. 請依照範例程式，寫出一支程式可以不斷的讀取溫度，直到抵達指定的溫度(CPU 溫度到達 35 度的擷圖, 如果沒有辦法到 35 度, 請自行選擇一個溫度並且附上到選擇溫度的擷圖)

(1)結果(因溫度一直很高故選擇 43 度)

```
s406410114 > (ssh) lonux > ~/OS_HW/hw0B.CPU_temperature > ./waitTemp
target = 43
#=01 sec 🌞 🔥 🌞 🔥 43.25 °C
#=02 sec 🌞 🔥 🌞 🔥 43.25 °C
#=03 sec 🌞 🔥 🌞 🔥 43.25 °C
#=04 sec 🌞 🔥 🌞 🔥 43.25 °C
temp 43.00°C
```

(2)修改

(i)修改讀取的資料路徑(用 strace sensors 取得)

```
temp1 = fopen("/sys/class/hwmon/hwmon4/temp1_input", "r");
assert(temp1 != NULL);
temp2 = fopen("/sys/class/hwmon/hwmon4/temp2_input", "r");
assert(temp2 != NULL);
temp3 = fopen("/sys/class/hwmon/hwmon4/temp3_input", "r");
assert(temp3 != NULL);
temp4 = fopen("/sys/class/hwmon/hwmon4/temp4_input", "r");
```

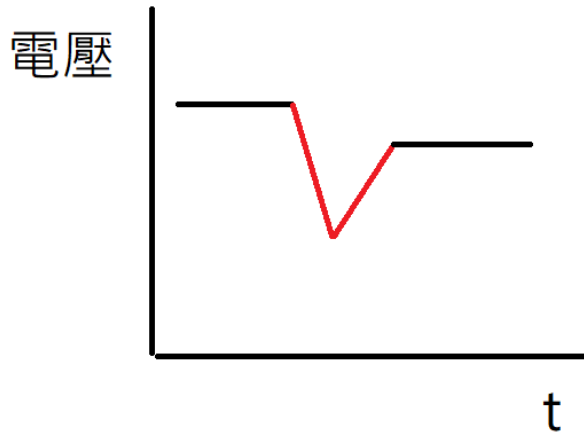
(ii)因溫度一直過高，故選擇 43 度

```
int main(int argc, char **argv) {
    //預設值等到 CPU 的溫度降到 35 度
    int tempture=43;
```

謝謝老師！謝謝助教！

//youtube 筆記

1. 若溫度靠近安全溫度(68 度)，會把頻率下降降溫(2.6→2.5)
2. 如果還是不行，會調低電壓，造成頻率大量下降來降溫(2.6→2.0)
3. 調整電壓需要讓 CPU 暫停一下，使用者會感受到性能明顯掉落，因電壓要等到穩定才可以運算，紅色區間 CPU 無法運算



4. 溫度適用類推的不一定很準

```
$ strace sensors #system_call trace
$ cd /sys/class/hwmon/hwmon1
$ cat name #看詳細資訊
$ cat templ_label #看 Label 資訊
$ googler K10temp #google 文字搜尋
```

//code

```
#define _GNU_SOURCE
#include <stdio.h>
#include <pthread.h>
#include <stdatomic.h>
#include <string.h>
#include <signal.h>
#include <unistd.h>
#include <stdlib.h>
#include <sched.h>
#include <stdio.h>
#include <string.h>
#include <sys/syscall.h>
#include <assert.h>
#include <string.h>
#include <stdbool.h>
int numCPU = -1;
```

```

struct CpuInfo {
    float mhz;
};

FILE* cpuinfo;
struct CpuInfo* cpuinfoArray;

int temp[4];
FILE* temp1, *temp2, *temp3, *temp4;

//底下這程式碼是印出執行頻率，我印象當中，底下程式碼是錯誤的
//必須把檔案關掉再打開，這樣才會顯示真正的頻率
void printFreq() {
    char buffer[1000];
    fseek(cpuinfo, 0, SEEK_SET);
    while(1) {
        int cpuid;
        float freq;
        if (fgets(buffer, 1000, cpuinfo) == NULL) break;
        char *typeStr = strtok(buffer, ":");
        if (strcmp("cpu MHz", typeStr) == 0) {
            char* temp = strtok(NULL, ":");
            sscanf(temp, "%f", &freq);
            if (cpuid == -1) {
                fprintf(stderr, "ERROR: cannot read temp correctly\n");
                break;
            }
            cpuinfoArray[cpuid].mhz = freq;
            cpuid = -1;
            //printf("MHZ = %s\n", temp);
        }
        if (strcmp("processor", typeStr) == 0) {
            char* cpuidstr = strtok(NULL, ":");
            sscanf(cpuidstr, "%d", &cpuid);
        }
    }
    float totalFreq = 0;
    for (int i = 0; i < numCPU; i++) {
        totalFreq += cpuinfoArray[i].mhz;
    }
    printf("freq, %.2f, ", totalFreq/numCPU);
}

```

```

}

//底下這程式碼會印出所有 CPU 的溫度
int printTemp() {
    int eatReturn;
    int tempArray[4];

    //重新打開讀取一次 要修改目錄 每台機器可能不一樣
    temp1 = fopen("/sys/class/hwmon/hwmon1/temp1_input", "r");
    assert(temp1 != NULL);
    temp2 = fopen("/sys/class/hwmon/hwmon2/temp1_input", "r");
    assert(temp2 != NULL);
    temp3 = fopen("/sys/class/hwmon/hwmon3/temp1_input", "r");
    assert(temp3 != NULL);
    temp4 = fopen("/sys/class/hwmon/hwmon4/temp1_input", "r");
    assert(temp4 != NULL);

    for (int i=0; i<4; i++)
        tempArray[i] = -1;
    //int eatReturn;
    //fseek 不一定要做
    fseek(temp1, 0, SEEK_SET);
    eatReturn=fscanf(temp1, "%d", &tempArray[0]);
    fseek(temp2, 0, SEEK_SET);
    eatReturn=fscanf(temp2, "%d", &tempArray[1]);
    fseek(temp3, 0, SEEK_SET);
    eatReturn=fscanf(temp3, "%d", &tempArray[2]);
    fseek(temp4, 0, SEEK_SET);
    eatReturn=fscanf(temp4, "%d", &tempArray[3]);

    fclose(temp1); fclose(temp2); fclose(temp3); fclose(temp4); //關閉

    int totalTemp=0;
    for (int i=0; i<4; i++)
        totalTemp += tempArray[i];
    //printf("temp, %d, ", totalTemp/4);
    //printf("%d \n", totalTemp);
    int avgTemp = totalTemp/4;
    return avgTemp;
}

```

```

atomic_int stop = 0;

long timespec2nano(struct timespec ts) {
    return ts.tv_sec * 1000000000 + ts.tv_nsec;
}

char* exename;

int main(int argc, char **argv) {
    //預設值等到 CPU 的溫度降到 35 度
    int tempture=35;
    //使用者可以在參數列輸入溫度
    if (argc == 2) {
        sscanf(argv[1], "%d", &tempture);
    }
    fprintf(stderr, "target = %d\n", tempture);
    //授課老師自己稍微看一下，這個溫度要乘上 1000 才是真正的溫度
    tempture = tempture * 1000;
    //底下的程式碼是我以前寫的，留給大家做參考
    //cpuinfo = fopen("/proc/cpuinfo", "r");
    //fprintf(stderr, "wait32, step 1...\n");
    //temp1 = fopen("/sys/class/hwmon/hwmon1/temp1_input", "r");
    //assert(temp1 != NULL);
    //temp2 = fopen("/sys/class/hwmon/hwmon2/temp1_input", "r");
    //assert(temp2 != NULL);
    //temp3 = fopen("/sys/class/hwmon/hwmon3/temp1_input", "r");
    //assert(temp3 != NULL);
    //temp4 = fopen("/sys/class/hwmon/hwmon4/temp1_input", "r");
    //assert(temp4 != NULL);
    //cpuinfoArray = (struct CpuInfo*)malloc(sizeof(struct CpuInfo) * numCPU);

    //一個迴圈，每隔一秒讀取 CPU 的溫度，直到溫度比設定的還要低
    int temp;
    int try=1;
    while((temp = printTemp()) > tempture) {
        fprintf(stderr, "#=%02d sec 🌀 🌀 🌀 🌀 %.2f °C\n", try,
((float)temp)/1000);
        try++;
        sleep(1);
    }
}

```

```
}  
printf("temp %.2f°C, \n", ((float)temp)/1000);  
exit(0);  
}
```