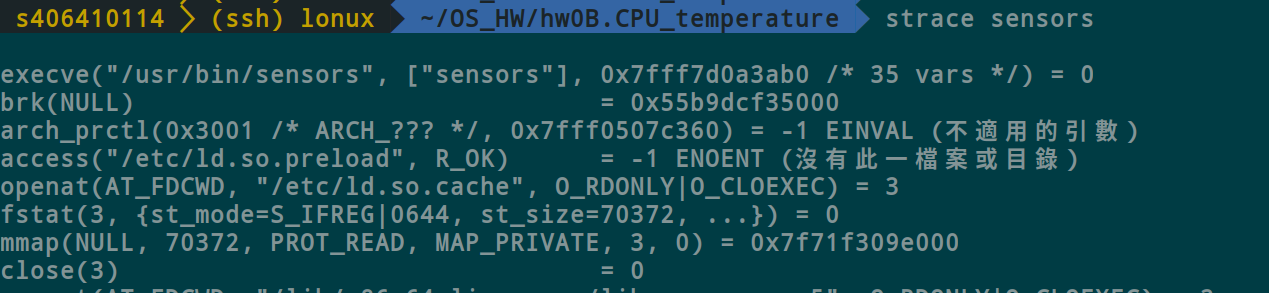
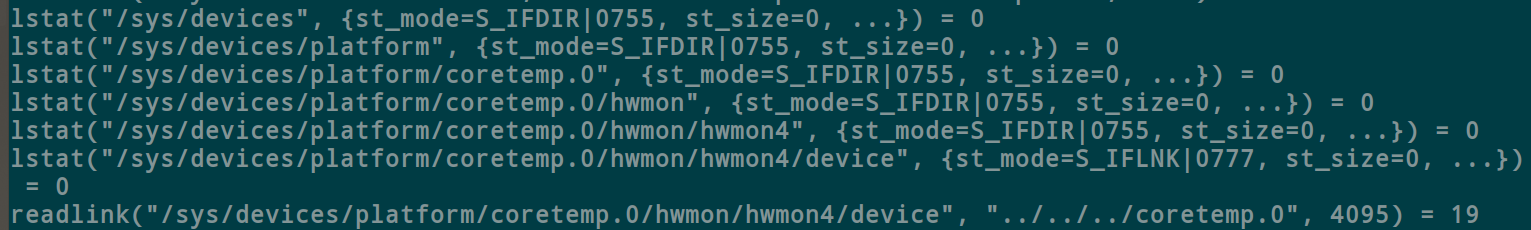
hw0B.CPU temperature

1. 使用strace在你的處理器上追蹤sensors這個指令

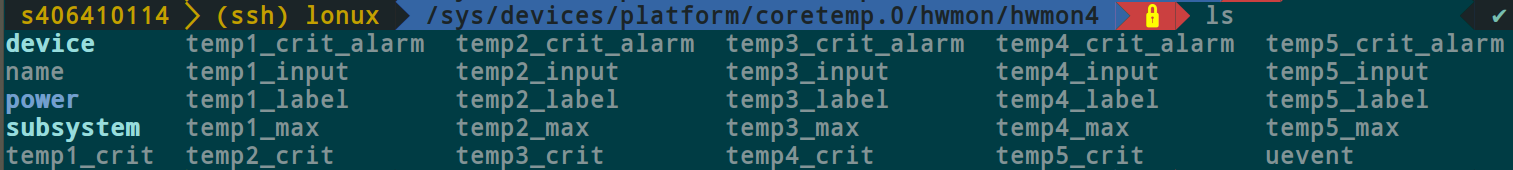


1. 列出sensors從哪邊讀取CPU的溫度(使用 strace trace sensors 如何讀取 CPU溫度擷圖 )

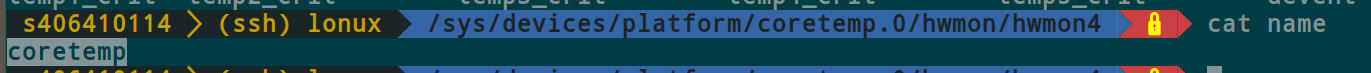
(1)不斷進入此目錄去讀取 ( /sys/devices/platform/coretemp.0/hwmon/hwmon4 )

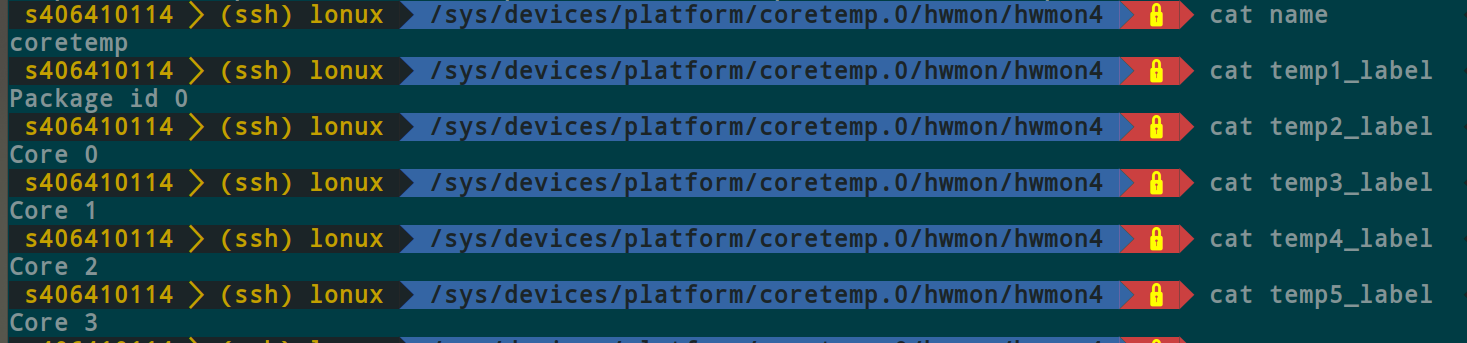


(2) cd /sys/devices/platform/coretemp.0/hwmon/hwmon4 ls出檔案



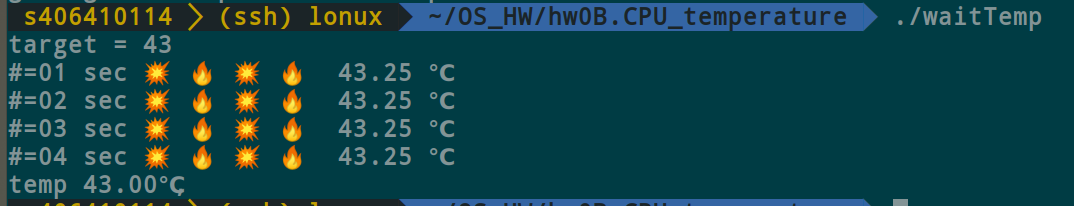
(3)cat name 出現coretemp



(4)cat temp\_label 可以發現各個CPU被標記了不同代號

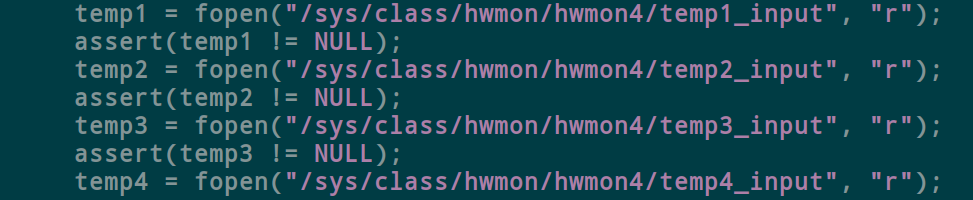
3.請依照範例程式，寫出一支程式可以不斷的讀取溫度，直到抵達指定的溫度(CPU溫度到達35度的擷圖,如果沒有辦法到35度,請自行選擇一個溫度並且附上到選擇溫度的擷圖)

(1)結果(因溫度一直很高故選擇43度)

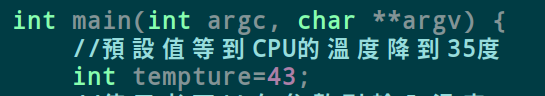


(2)修改

(i)修改讀取的資料路徑(用strace sensors取得)



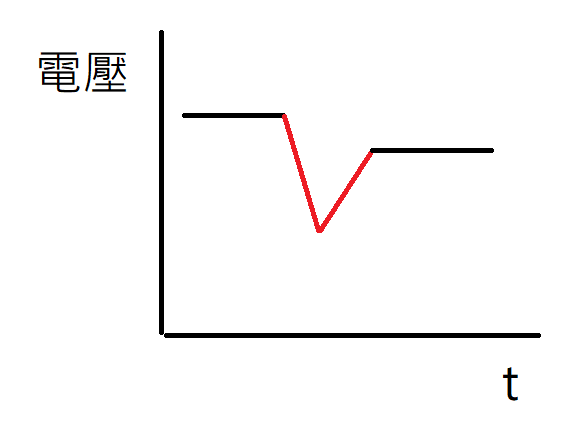
(ii)因溫度一直過高，故選擇43度



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



1. 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 temp1\_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 printFeq() {

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 ℃\n", try, ((float)temp)/1000);

try++;

sleep(1);

}

printf("temp %.2f℃, \n", ((float)temp)/1000);

exit(0);

}