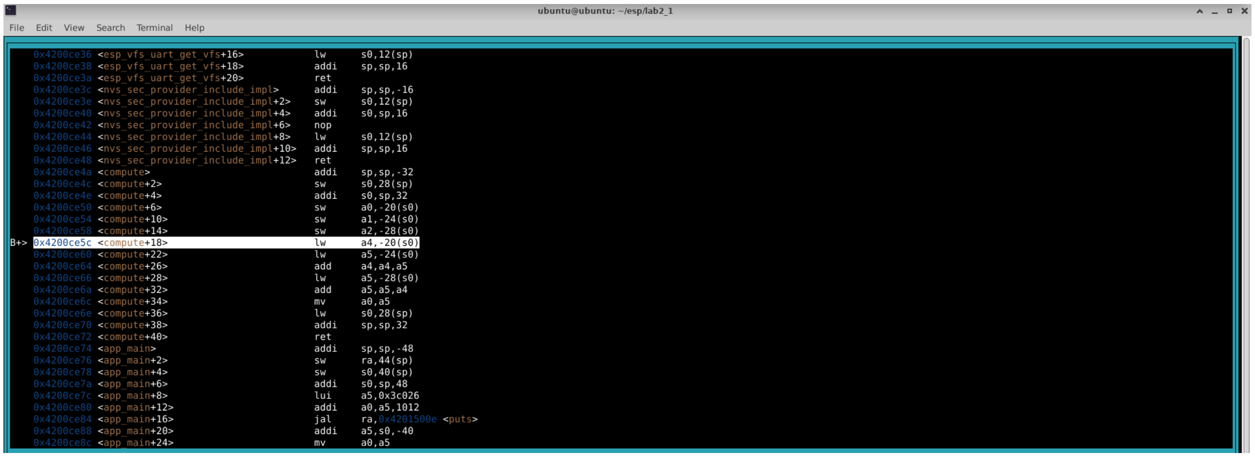
Xavier Thompson
10/19/23

CSE121 Lab Report 2

Lab 2.1

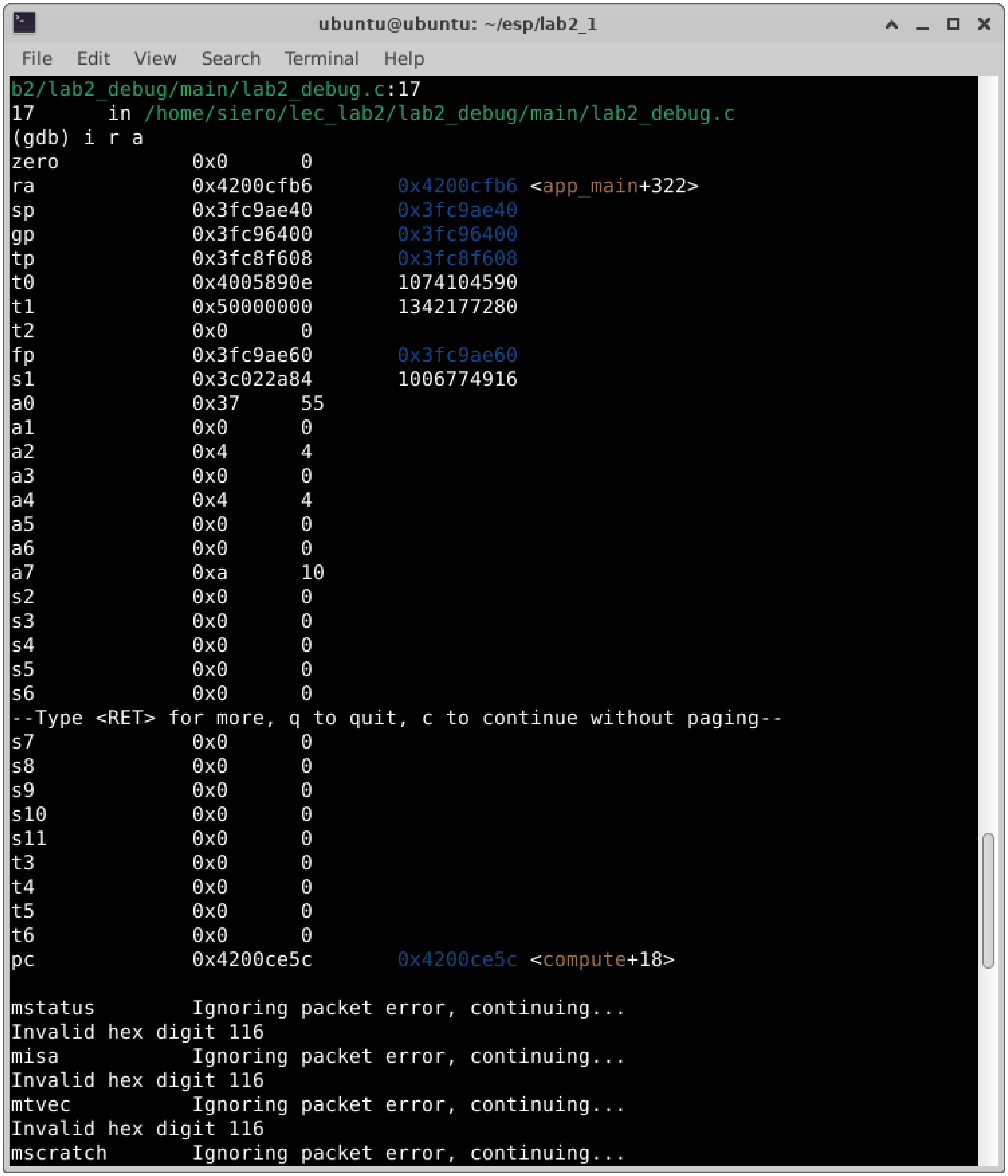
Compute 1st argument is 55 and is passed via a0
Compute 2nd argument is 0 and is passed via a1
Compute 3rd argument is 4 and is passed via a2
Return value is 59 and is returned via 0x4200ce72
The entry point of the "compute" function is at address 0x4200ce4a
The instruction that returns the value is at address 0x4200ce72

The screenshot below shows the addresses for lab2_1.


ubuntu@ubuntu: ~/esp/lab2_1
File Edit View Search Terminal Help

```
    0x4200ce36 <esp_vfs_uart_get_vfs+16>           lw      s0,12(sp)
    0x4200ce38 <esp_vfs_uart_get_vfs+18>           addi    sp,sp,16
    0x4200ce3a <esp_vfs_uart_get_vfs+20>           ret
    0x4200ce3c <nvs_sec_provider_include_impl>     addi    sp,sp,-16
    0x4200ce3e <nvs_sec_provider_include_impl+2>   sw      s0,12(sp)
    0x4200ce40 <nvs_sec_provider_include_impl+4>   addi    s0,sp,16
    0x4200ce42 <nvs_sec_provider_include_impl+6>   nop
    0x4200ce44 <nvs_sec_provider_include_impl+8>   lw      s0,12(sp)
    0x4200ce46 <nvs_sec_provider_include_impl+10>  addi    sp,sp,16
    0x4200ce48 <nvs_sec_provider_include_impl+12>  ret
    0x4200ce4a <compute>                           addi    sp,sp,-32
    0x4200ce4c <compute+2>                         sw      s0,28(sp)
    0x4200ce4e <compute+4>                         addi    s0,sp,32
    0x4200ce50 <compute+6>                         sw      a0,-20(s0)
    0x4200ce54 <compute+10>                        sw      a1,-24(s0)
    0x4200ce58 <compute+14>                        sw      a2,-28(s0)
B+> 0x4200ce5c <compute+18>                        lw      a4,-20(s0)
    0x4200ce60 <compute+22>                        lw      a5,-24(s0)
    0x4200ce64 <compute+26>                        add     a4,a4,a5
    0x4200ce66 <compute+28>                        lw      a5,-28(s0)
    0x4200ce6a <compute+32>                        add     a5,a5,a4
    0x4200ce6c <compute+34>                        mv      a0,a5
    0x4200ce6e <compute+36>                        lw      s0,28(sp)
    0x4200ce70 <compute+38>                        addi    sp,sp,32
    0x4200ce72 <compute+40>                        ret
    0x4200ce74 <app_main>                          addi    sp,sp,-48
    0x4200ce76 <app_main+2>                        sw      ra,44(sp)
    0x4200ce78 <app_main+4>                        sw      s0,40(sp)
    0x4200ce7a <app_main+6>                        addi    s0,sp,48
    0x4200ce7c <app_main+8>                        lui     a5,0x3c026
    0x4200ce80 <app_main+12>                       addi    a0,a5,1012
    0x4200ce84 <app_main+16>                       jal     ra,0x4201500e <puts>
    0x4200ce88 <app_main+20>                       addi    a5,s0,-40
    0x4200ce8c <app_main+24>                       mv      a0,a5
```

The screenshot below shows the addresses for lab2_1.



```
b2/lab2_debug/main/lab2_debug.c:17
17      in /home/siero/lec_lab2/lab2_debug/main/lab2_debug.c
(gdb) i r a
zero           0x0      0
ra             0x4200cfb6      0x4200cfb6 <app_main+322>
sp             0x3fc9ae40      0x3fc9ae40
gp             0x3fc96400      0x3fc96400
tp             0x3fc8f608      0x3fc8f608
t0             0x4005890e      1074104590
t1             0x50000000      1342177280
t2             0x0      0
fp             0x3fc9ae60      0x3fc9ae60
s1             0x3c022a84      1006774916
a0             0x37     55
a1             0x0      0
a2             0x4      4
a3             0x0      0
a4             0x4      4
a5             0x0      0
a6             0x0      0
a7             0xa      10
s2             0x0      0
s3             0x0      0
s4             0x0      0
s5             0x0      0
s6             0x0      0
--Type <RET> for more, q to quit, c to continue without paging--
s7             0x0      0
s8             0x0      0
s9             0x0      0
s10            0x0      0
s11            0x0      0
t3             0x0      0
t4             0x0      0
t5             0x0      0
t6             0x0      0
pc             0x4200ce5c      0x4200ce5c <compute+18>

mstatus        Ignoring packet error, continuing...
Invalid hex digit 116
misa           Ignoring packet error, continuing...
Invalid hex digit 116
mtvec          Ignoring packet error, continuing...
Invalid hex digit 116
mscratch       Ignoring packet error, continuing...
```

While trying to get the debugger to work, I also used the following links to fix my bugs.

Problem 1: Configuring JTAG rules on esp32

https://docs.espressif.com/projects/esp-idf/en/latest/esp32c3/api-guides/jtag-debugging/configure-builtin-jtag.html

Problem 2: Run openocd and flash binary onto chip
https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/jtag-debugging/index.html

Problem 3: Convert .elf file to binary image
https://docs.espressif.com/projects/esptool/en/latest/esp32s3/esptool/basic-commands.html#convert-elf-to-binary-elf2image

Problem 4: Openocd process does not have permissions to access the JTAG/serial device

ChatGPT Answer: The error message "OpenOCD process does not have permissions to access the USB JTAG/serial device" typically indicates that the OpenOCD process doesn't have the necessary permissions to access the JTAG/serial device, which is commonly connected via USB. To resolve this issue, you can follow these steps:

1. **Add User to the Dialout Group**:
   The most common reason for this error is that the user running OpenOCD does not have the necessary permissions to access the USB device. To fix this, you can add your user to the "dialout" group, which often has the required permissions.
   Run the following command to add your user to the "dialout" group:
   bash

```
sudo usermod -a -G dialout your_username
```

Replace `your_username` with your actual username.
Log Out and Log In:
After adding your user to the "dialout" group, you may need to log out and then log back in for the group changes to take effect.
Check udev Rules:
Ensure that the USB JTAG/serial device has the correct udev rules. These rules define the permissions and access rights for USB devices. Check whether there are specific udev rules for your device and whether they are correctly configured.
Reload udev Rules:
After modifying udev rules, you may need to reload the rules to apply the changes. Run the following command to reload udev rules:
bash

```
sudo udevadm control --reload-rules
```

Unplug and Replug the Device:
Sometimes, simply unplugging and replugging the USB JTAG/serial device can resolve the issue. This can help in cases where the device was detected with incorrect permissions or if the device wasn't recognized when you added your user to the "dialout" group.
Run OpenOCD with sudo:
If you are still facing permission issues, as a last resort, you can run OpenOCD with sudo. However, using sudo should be used with caution because it grants the program root privileges, and it might not be recommended for security reasons.
For example:
bash

```
sudo openocd -f interface/your_interface.cfg -f
target/your_target.cfg
```

1. Replace `your_interface.cfg` and `your_target.cfg` with the appropriate configuration files for your setup.

Please note that running OpenOCD with sudo should be a temporary solution, and you should try to resolve the permissions issues using the previous steps. It's essential to balance the need for device access with system security and permissions best practices.

Lab2.2

My code for Lab 2.2 was adapted from the following code provided by chatGPT.

Prompt
Build a C program that uses SHTC3 sensor on esp32c3 to display the temperature and humidity every 2 seconds using I2C protocol

Answer:

```
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "driver/i2c.h"

#define I2C_MASTER_SCL_IO        22
#define I2C_MASTER_SDA_IO        21
#define I2C_MASTER_NUM           I2C_NUM_0
```

```
#define I2C_MASTER_FREQ_HZ   100000  // I2C frequency

#define SHTC3_ADDR          0x70  // SHTC3 I2C address

void app_main()
{
        i2c_config_t conf;
        conf.mode = I2C_MODE_MASTER;
        conf.sda_io_num = I2C_MASTER_SDA_IO;
        conf.sda_pullup_en = GPIO_PULLUP_ENABLE;
        conf.scl_io_num = I2C_MASTER_SCL_IO;
        conf.scl_pullup_en = GPIO_PULLUP_ENABLE;
        conf.master.clk_speed = I2C_MASTER_FREQ_HZ;
        i2c_param_config(I2C_MASTER_NUM, &conf);
        i2c_driver_install(I2C_MASTER_NUM, I2C_MODE_MASTER, 0, 0, 0);

        uint8_t data[6];
        int16_t temperature;
        int16_t humidity;

        while (1)
        {
        i2c_cmd_handle_t cmd = i2c_cmd_link_create();
        i2c_master_start(cmd);
        i2c_master_write_byte(cmd, (SHTC3_ADDR << 1) | I2C_MASTER_READ, true);
        i2c_master_read(cmd, data, 6, I2C_MASTER_LAST_NACK);
        i2c_master_stop(cmd);
        esp_err_t ret = i2c_master_cmd_begin(I2C_MASTER_NUM, cmd, 1000 /
portTICK_RATE_MS);
        i2c_cmd_link_delete(cmd);

        if (ret == ESP_OK)
        {
        temperature = (data[0] << 8) | data[1];
        humidity = (data[3] << 8) | data[4];

        float temp_degC = -45 + 175 * (temperature / 65535.0);
        float hum_percent = 100 * (humidity / 65535.0);

        printf("Temperature: %.2f°C, Humidity: %.2f%%\n", temp_degC, hum_percent);
```

```
    }
    else
    {
    printf("Failed to read data from SHTC3 sensor\n");
    }

    vTaskDelay(2000 / portTICK_PERIOD_MS);  // Wait for 2 seconds
    }
}
```

##

For my code, I set the SDA and SCL I/O pins to be 10 and 8 because I was having issues with the ones provided by chatGPT. I also changed the conf definition to a dictionary because I was getting an invalid flags/frequency error. I got the idea to change it to a dictionary from this link: https://esp32.com/viewtopic.php?t=21030
After making those changes, I just added an equation to set the Fahrenheit temperature to 1.8*degC + 32 and formatted the printing using ESP_LOGI.