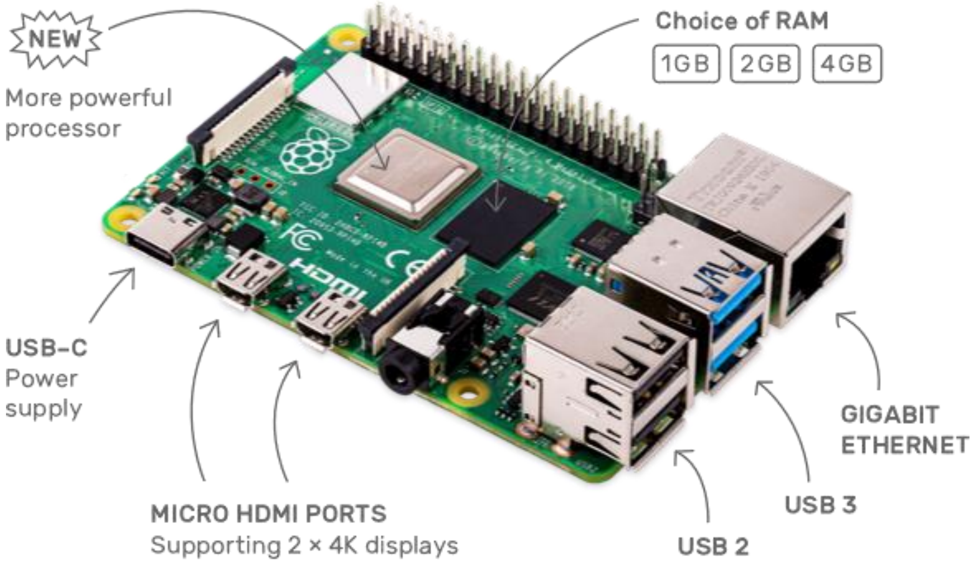
EECS 388 Lab #7

Intro to Embedded Linux

In this lab, you will setup your account on your Raspberry Pi 4, and be familiarized with the system.



Unlike the HiFive1 micro-controller you used, the Raspberry Pi 4 in front of you is essentially a small PC, which runs a general purpose operating system, Linux, complete with a desktop environment. You can basically do anything that you would expect to do on a Linux based PC. On the Raspberry Pi 4, using its computing power, you will later run a deep learning model for vision based real-time control.

Part 0: Setup your account in the Raspberry Pi 4.

Before you do any of such fancy things, you first need to log in to the Pi 4 account.

Seven user ID's are already created at the Pi4.

Login ID's: lab1 lab2 lab3 lab4 lab5 lab6 lab7

You need to put the password same as the login ID to get access your Pi 4 account.

For example, if you use lab4 as user ID, you have to use lab4 as the password for login.

You must use the same ID to login to your Pi4 account for rest of the labs.

Don't change the password of your Pi4 account.

Part 1: Getting familiar with the commonly used Linux tools

In this lab, the main goal is to get familiar with the system.

Task 1.1 Hello World in C

Use your favorite editor (vi, emacs, or anything), edit the following hello world in C and python

```
$ vi hello.c
#include <stdio.h>
int main()
{
  printf("hello world!\n");
  return 0;
}
$ gcc hello.c
$ ./a.out
hello world!
```

Task 2. Hello World in Python

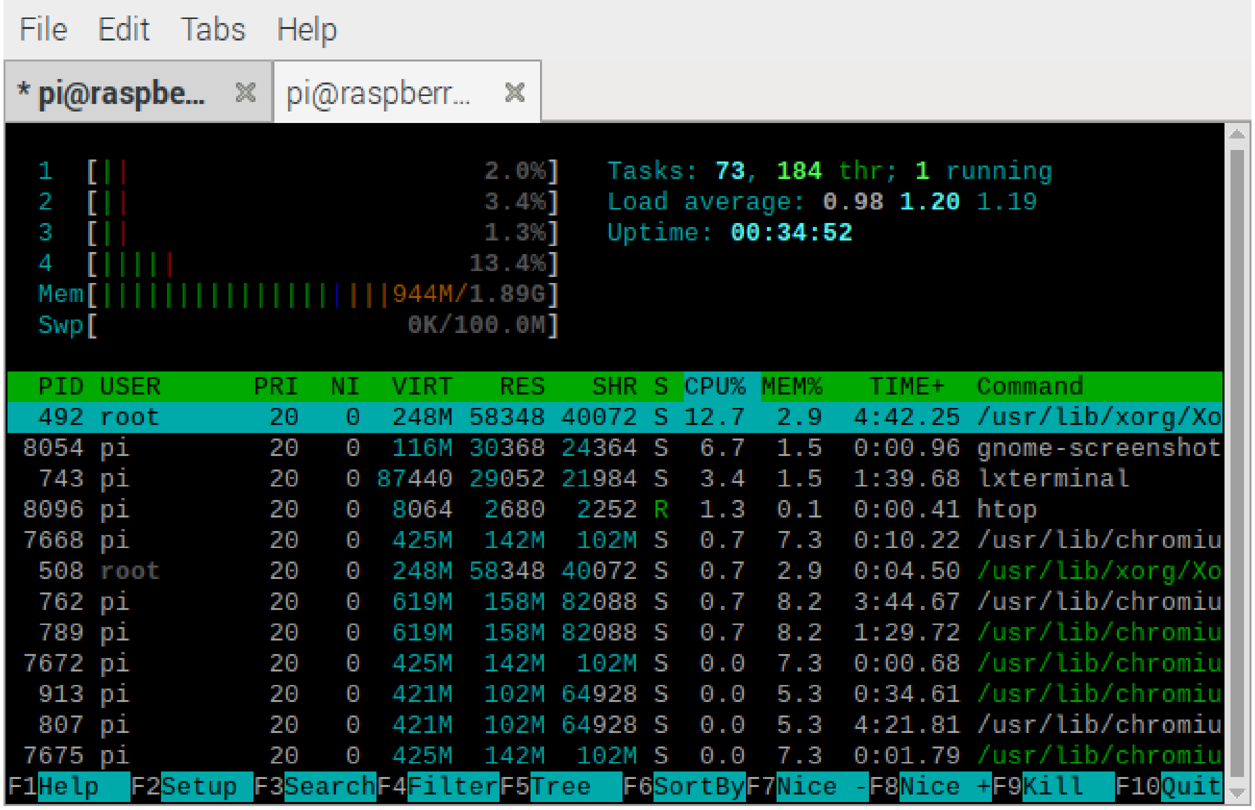
Do the same hello world with python as follows.

```
$ python
Python 2.7.16 (default, Apr  6 2019, 01:42:57)
[GCC 8.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Hello World!")
Hello World!
```

Task 3. System monitoring

Let’s check what programs are currently running on the pi 4 using `htop`. Note that Pi 4 has 4 cores and 2 GB of memory. You can visually see what programs are running on which CPU cores and how much they are being used.

```
$ htop
```



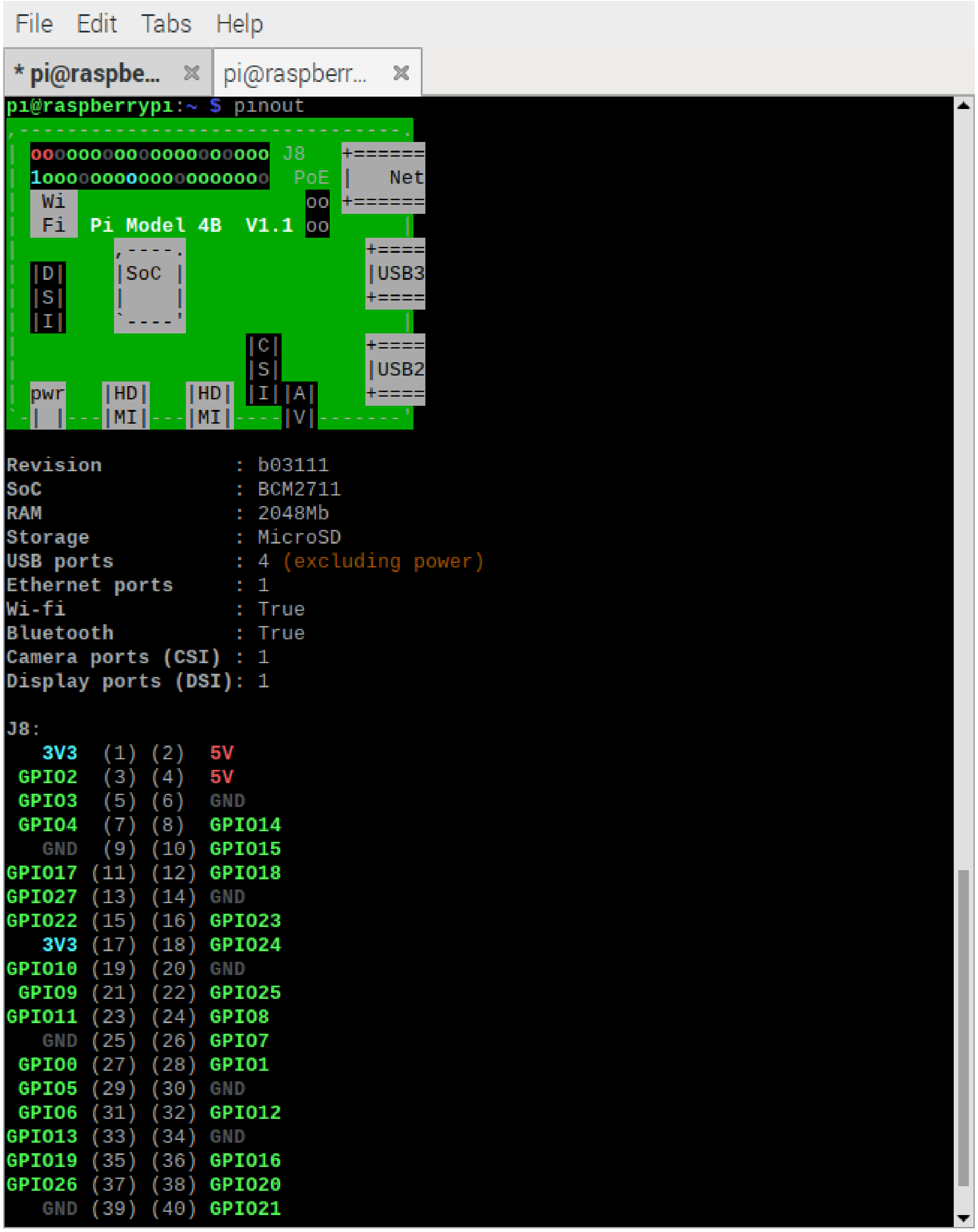
```
  1  [||                        2.0%]   Tasks: 73, 184 thr; 1 running
  2  [||                        3.4%]   Load average: 0.98 1.20 1.19
  3  [||                        1.3%]   Uptime: 00:34:52
  4  [|||||                    13.4%]
  Mem[|||||||||||||||||||944M/1.89G]
  Swp[                    0K/100.0M]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  492 root       20   0  248M 58348 40072 S 12.7  2.9  4:42.25 /usr/lib/xorg/Xo
 8054 pi         20   0  116M 30368 24364 S  6.7  1.5  0:00.96 gnome-screenshot
  743 pi         20   0 87440 29052 21984 S  3.4  1.5  1:39.68 lxterminal
 8096 pi         20   0  8064  2680  2252 R  1.3  0.1  0:00.41 htop
 7668 pi         20   0  425M  142M  102M S  0.7  7.3  0:10.22 /usr/lib/chromiu
  508 root       20   0  248M 58348 40072 S  0.7  2.9  0:04.50 /usr/lib/xorg/Xo
  762 pi         20   0  619M  158M 82088 S  0.7  8.2  3:44.67 /usr/lib/chromiu
  789 pi         20   0  619M  158M 82088 S  0.7  8.2  1:29.72 /usr/lib/chromiu
 7672 pi         20   0  425M  142M  102M S  0.0  7.3  0:00.68 /usr/lib/chromiu
  913 pi         20   0  421M  102M 64928 S  0.0  5.3  0:34.61 /usr/lib/chromiu
  807 pi         20   0  421M  102M 64928 S  0.0  5.3  4:21.81 /usr/lib/chromiu
 7675 pi         20   0  425M  142M  102M S  0.0  7.3  0:01.79 /usr/lib/chromiu
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Now, open another terminal and run the following simple C program and see how it is shown in the htop screen.

```
$ vi cpuhog.c
int main()
{
  while(1);
  return 0;
}
$ gcc cpuhog.c -o cpuhog
$ ./cpuhog
```

There are a couple of very useful Raspberry Pi specific tools. The pinout is such a program. You may need the tool in the future when you connect the Pi 4 to other sensors and the HiFive 1.

```
$ pinout
```

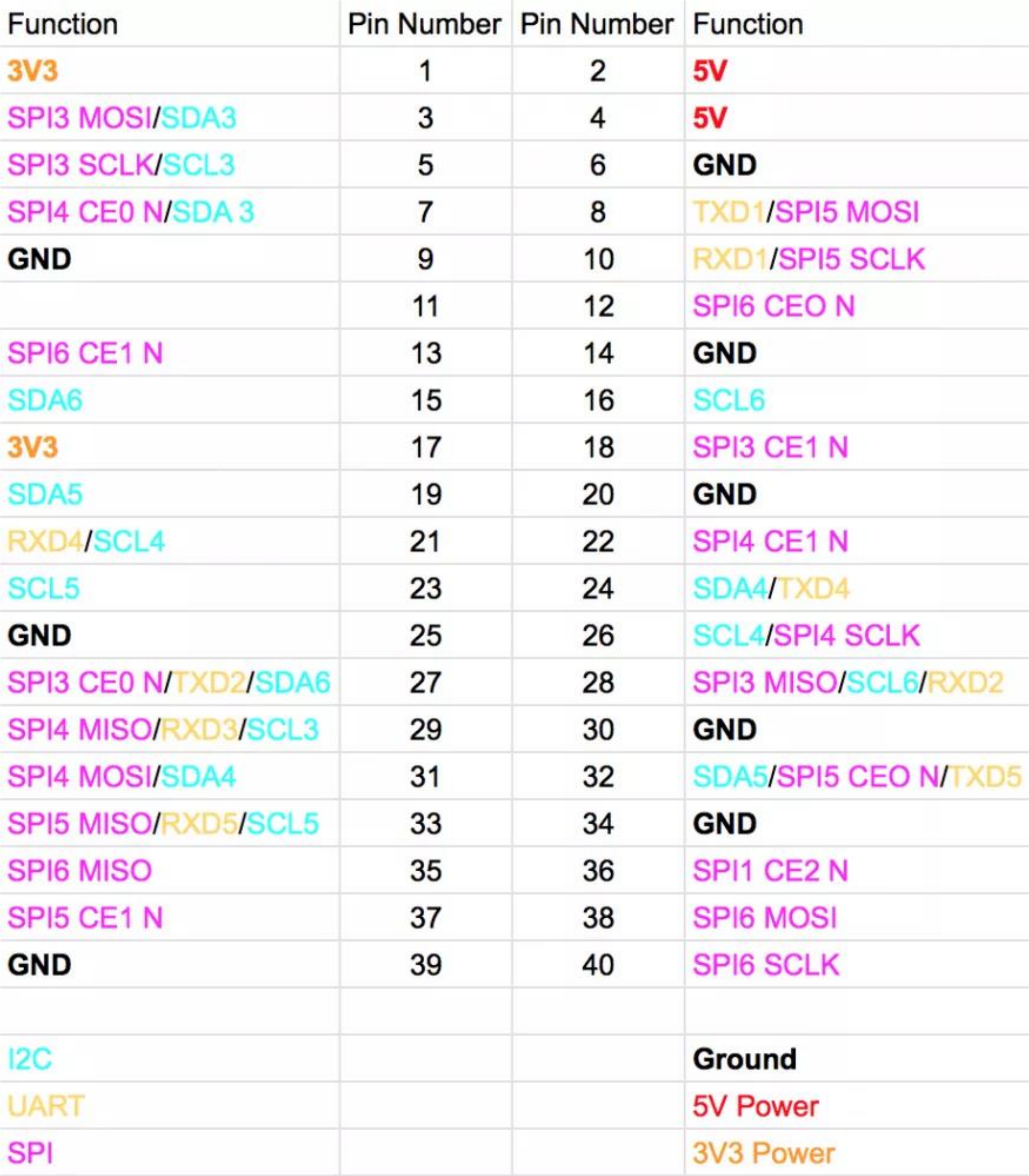


```
pi@raspberrypi:~ $ pinout
,--------------------------------.
| oooooooooooooooooooo J8   +======
| 1ooooooooooooooooooo  PoE |   Net
|  Wi                    oo +======
|  Fi  Pi Model 4B  V1.1 oo      |
|        ,----.               +====
| |D|    |SoC |               |USB3
| |S|    |    |               +====
| |I|    `----'                   |
|                   |C|       +====
|                   |S|       |USB2
| pwr   |HD|   |HD| |I||A|    +====
`-| |---|MI|---|MI|----|V|-------'

Revision           : b03111
SoC                : BCM2711
RAM                : 2048Mb
Storage            : MicroSD
USB ports          : 4 (excluding power)
Ethernet ports     : 1
Wi-fi              : True
Bluetooth          : True
Camera ports (CSI) : 1
Display ports (DSI): 1

J8:
   3V3  (1) (2)  5V
 GPIO2  (3) (4)  5V
 GPIO3  (5) (6)  GND
 GPIO4  (7) (8)  GPIO14
   GND  (9) (10) GPIO15
GPIO17 (11) (12) GPIO18
GPIO27 (13) (14) GND
GPIO22 (15) (16) GPIO23
   3V3 (17) (18) GPIO24
GPIO10 (19) (20) GND
 GPIO9 (21) (22) GPIO25
GPIO11 (23) (24) GPIO8
   GND (25) (26) GPIO7
 GPIO0 (27) (28) GPIO1
 GPIO5 (29) (30) GND
 GPIO6 (31) (32) GPIO12
GPIO13 (33) (34) GND
GPIO19 (35) (36) GPIO16
GPIO26 (37) (38) GPIO20
   GND (39) (40) GPIO21
```

There will be no Demo / submission for this lab. This lab is designed for Raspberry Pi introduction.

Appendix

GPIO mapping of Pi 4.



Function	Pin Number	Pin Number	Function
3V3	1	2	5V
SPI3 MOSI/SDA3	3	4	5V
SPI3 SCLK/SCL3	5	6	GND
SPI4 CE0 N/SDA 3	7	8	TXD1/SPI5 MOSI
GND	9	10	RXD1/SPI5 SCLK
	11	12	SPI6 CEO N
SPI6 CE1 N	13	14	GND
SDA6	15	16	SCL6
3V3	17	18	SPI3 CE1 N
SDA5	19	20	GND
RXD4/SCL4	21	22	SPI4 CE1 N
SCL5	23	24	SDA4/TXD4
GND	25	26	SCL4/SPI4 SCLK
SPI3 CE0 N/TXD2/SDA6	27	28	SPI3 MISO/SCL6/RXD2
SPI4 MISO/RXD3/SCL3	29	30	GND
SPI4 MOSI/SDA4	31	32	SDA5/SPI5 CEO N/TXD5
SPI5 MISO/RXD5/SCL5	33	34	GND
SPI6 MISO	35	36	SPI1 CE2 N
SPI5 CE1 N	37	38	SPI6 MOSI
GND	39	40	SPI6 SCLK
I2C			Ground
UART			5V Power
SPI			3V3 Power

Source: https://learn.pi-supply.com/make/raspberry-pi-4-pinout

Disclaimer: This document is edited on a Raspberry Pi 4 using chromium-browser.