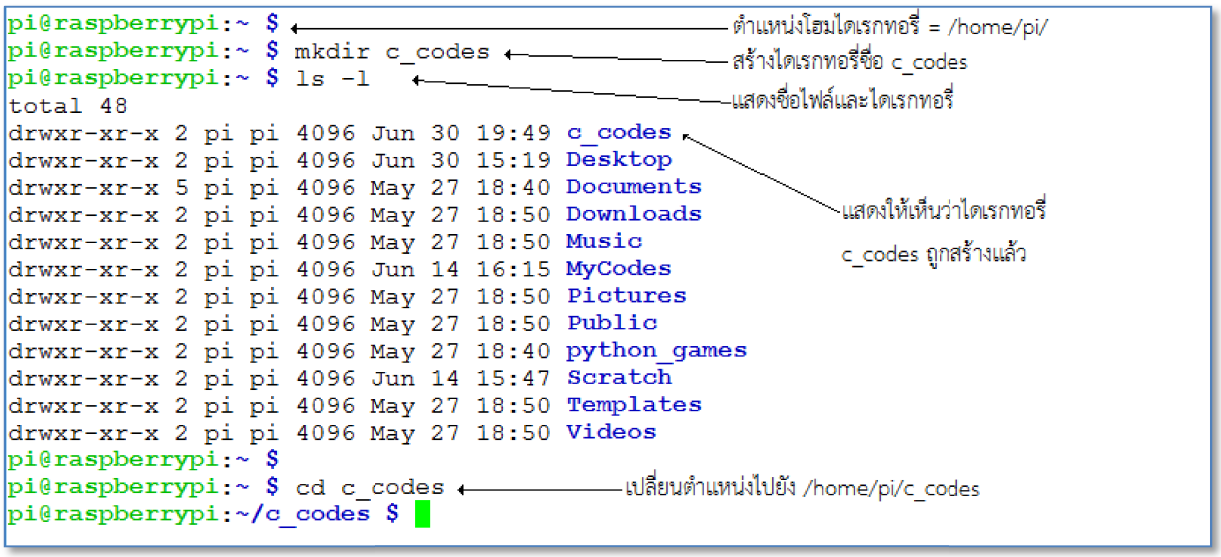
เขียน คอมไพล์ รัน โปรแกรมแรก - Hello world!

การหัดเขียนโปรแกรมภาษาใดก็แล้วแต่มักเริ่มต้นด้วยการสั่งให้แสดงข้อความที่ต้องการแสดง เช่น “Hello world!” (สวัสดีชาวโลก) เพื่อให้โปรแกรมที่เขียนตามเอกสารนี้จัดเก็บไว้อย่างเหมาะสมผู้เขียนจะสร้างไดเรกทอรี่ชื่อ c_codes ในโฮมไดเรกทอรี่เพื่อใช้เก็บโปรแกรมที่เราเขียน และเข้าไปในไดเรกทอรี่ดังกล่าว

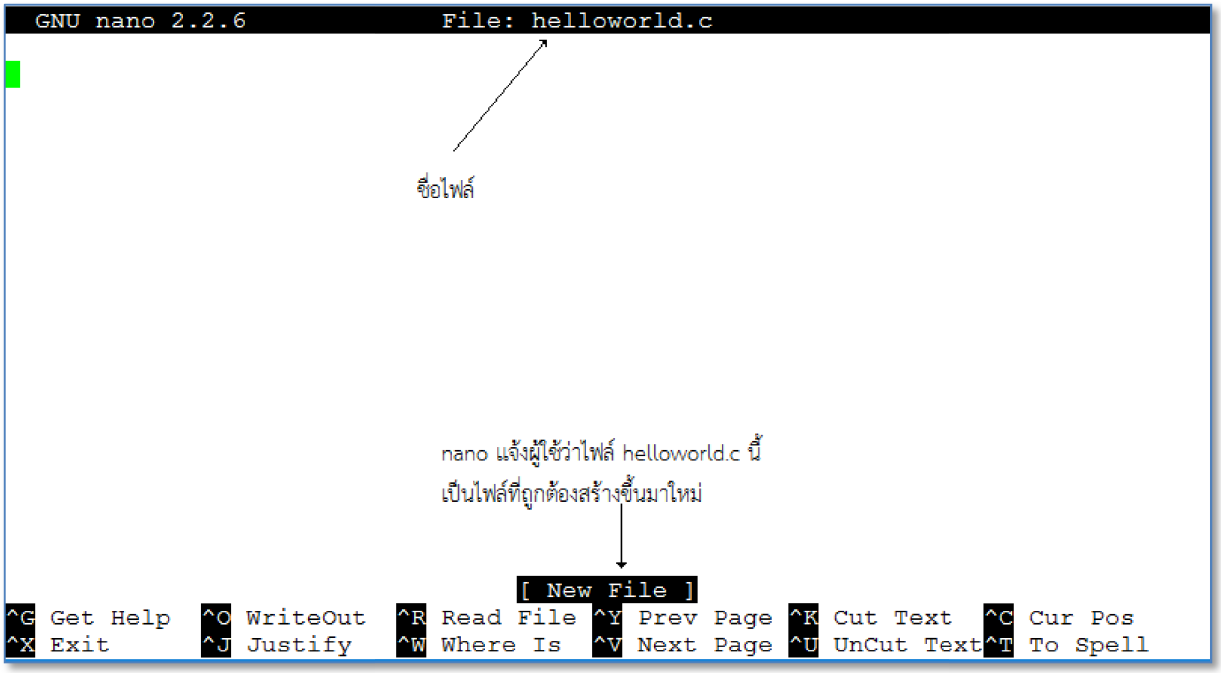


เขียนโปรแกรม

เมื่อเราอยู่ในตำแหน่ง /home/pi/c_code ตามภาพข้างบนแล้วทำการสร้างไฟล์ ชื่อว่า “helloworld.c” ด้วยโปรแกรม nano ดังนี้

```
nano helloworld.c
```

มีความหมายว่าเรียกใช้โปรแกรม nano และเปิดไฟล์ชื่อ helloworld.c ด้วย ในกรณีที่ไฟล์นี้อยู่แล้ว nano จะแสดงข้อมูลเก่าในไฟล์ขึ้นมาแสดงเพื่อให้เราสามารถ เพิ่ม ลบ แก้ไขได้ แต่ในกรณีที่ไม่เคยมีไฟล์ชื่อดังกล่าวอยู่ nano จะให้เราเพิ่มข้อมูลใหม่ (ข้อมูลในที่นี้หมายถึงคำสั่งภาษาซี) อ้างอิงจากข้างต้นไดเรกทอรี่ c_code และ helloworld.c พึ่งจะถูกสร้าง nano จะแสดงข้อความว่า [New File] ดังภาพ

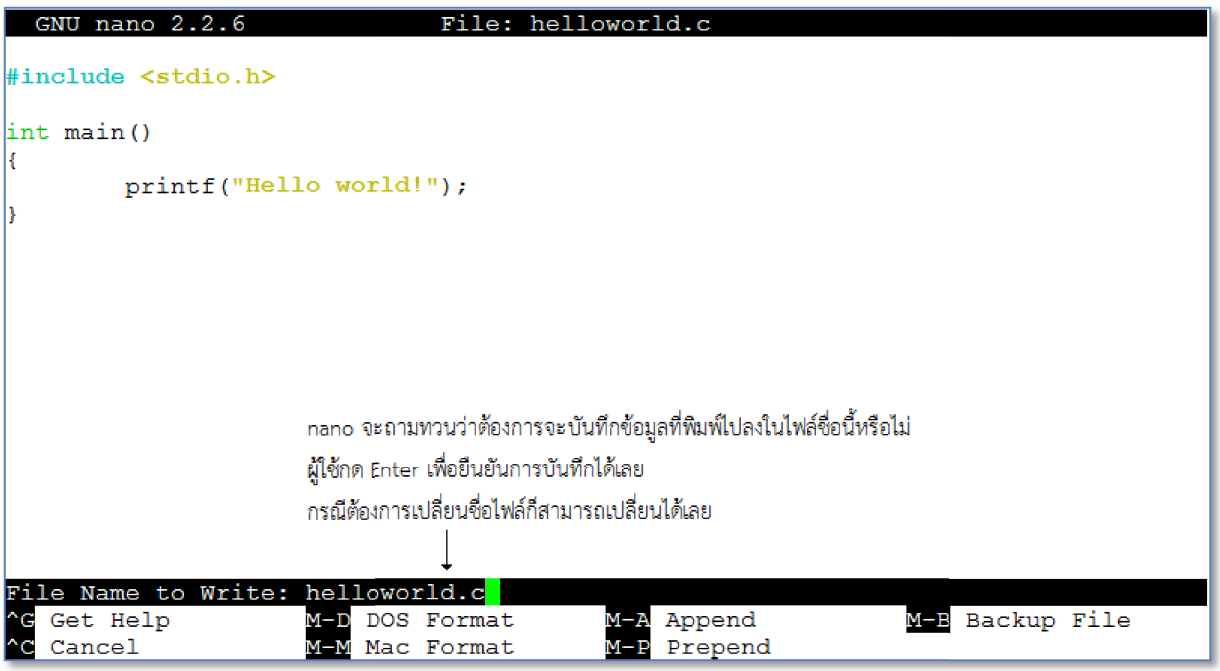


พิมพ์คำสั่งภาษาซีต่อไปนี้

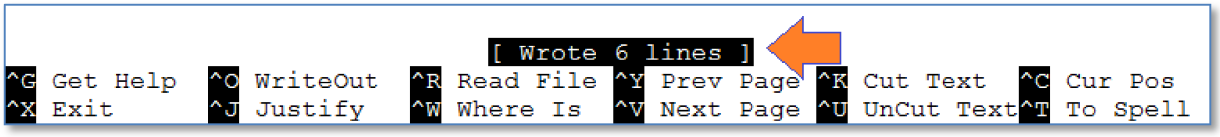
```
#include <stdio.h>

int main()
{
        printf("Hello world!");
}
```

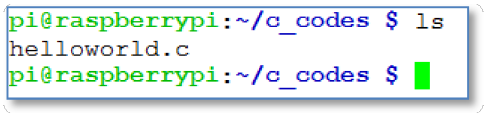
ตรวจสอบความถูกต้อง ทำการบันทึกด้วยปุ่ม Ctrl + o



แสดงสถานะว่าบันทึกแล้วทั้งสิ้นมี 6 บรรทัด



กด Ctrl +x ออกจาก nano ถือว่าเสร็จขั้นตอนการเขียนโค้ด เมื่อ ls จะพบว่ามีไฟล์ชื่อ helloworld.c



คอมไพล์โปรแกรม

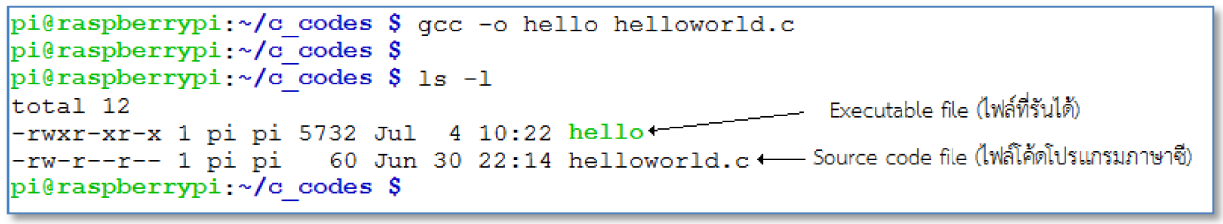
ต่อไปเราจะนำเอาโค้ดไปคอมไพล์ เพื่อให้ได้ไฟล์ที่สามารถใช้รันได้มาใช้งาน (executable file) รูปแบบการคอมไพล์ทำได้ดังนี้

```
gcc –o ชื่อไฟล์ที่ใช้รัน ชื่อไฟล์ภาษาซี.c
```

จากรูปแบบข้างต้น ตัวคอมไพเลอร์ชื่อว่า gcc จะนำเอาโค้ดภาษาซีที่ได้เขียนไว้ไปทำการคอมไพล์และลิงค์อ็อบเจกต์เข้าด้วยกัน แล้วแปลงให้เป็นไฟล์ที่สามารถรันได้ตามที่กำหนดไว้ –o เมื่อทราบรูปแบบการคอมไพล์แล้วลองทำการคอมไพล์จริงดังนี้

```
gcc –o hello helloworld.c
```

หมายความว่า คอมไพเลอร์ gcc ได้นำเอาไฟล์ภาษาซีชื่อ helloworld.c มาคอมไพล์ ลิงค์อ็อบเจ็กต์ แล้วนำมาสร้างเป็นไฟล์เอาท์พุต (-o) ที่สามารถใช้รันได้ไฟล์นั้นชื่อว่า hello



จากภาพด้านบนการคอมไพล์ประสบความสำเร็จเนื่องจากคอมไพเลอร์ไม่พบความผิดพลาดใดๆ ระหว่างคอมไพล์ เมื่อสั่ง ls -l พบว่ามีไฟล์เพิ่มขึ้นชื่อ hello และชื่อไฟล์เป็นสีเขียวหมายถึงเป็นไฟล์ที่สามารถรันหรือทำงาน(Executable file) ได้

รันโปรแกรม

เมื่อได้ Executable file มาแล้วสามารถเรียกใช้งานได้ตามรูปแบบดังนี้

```
ตำแหน่งที่อยู่ของไฟล์/ชื่อไฟล์
```

จากไฟล์ที่ได้ดำเนินการไว้สามารถเรียกใช้ดังนี้

```
/home/pi/c_codes/hello
```

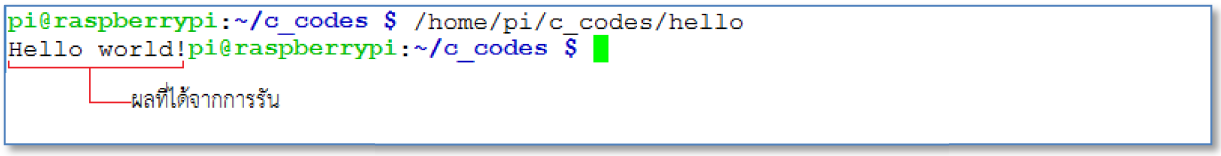
หรือ ตำแหน่งที่เราอยู่ในปัจจุบันคือ /home/pi/c_codes อยู่แล้วสามารถเรียกใช้ดังนี้ก็ได้

```
./hello
```

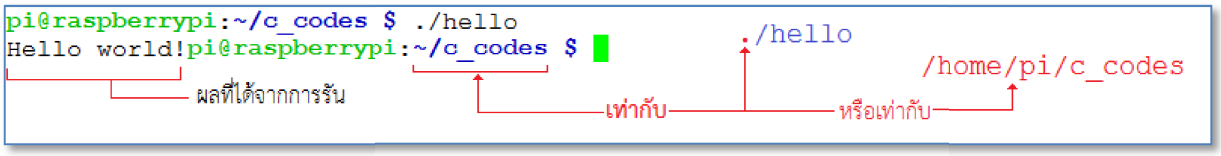
เครื่องหมายจุลภาค (.) ใช้ในการอ้างถึงตำแหน่งปัจจุบันที่อยู่

ผลการรัน

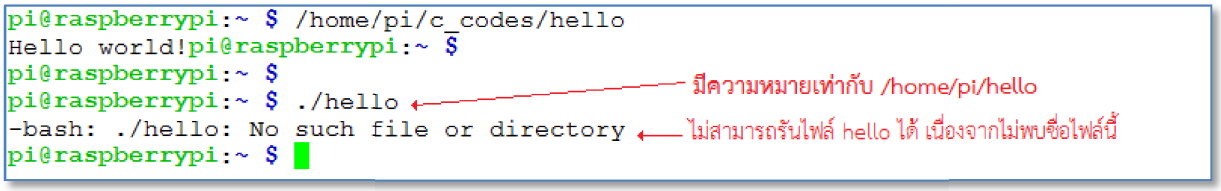
เมื่อทำการรันด้วยคำสั่ง /home/pi/c_codes/hello จะได้ผลดังนี้



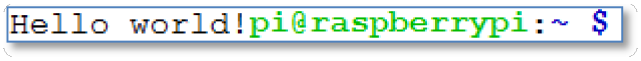
เมื่อทำการรันด้วยคำสั่ง ./hello จะได้ผลดังนี้



ผลของการรันทั้งสองคำสั่งรัน ได้ผลลัพธ์เหมือนกัน ต่างกันตรงที่การอ้างตำแหน่งที่อยู่ หรือ path ในการรันครั้งแรกจะอ้างอิงไฟล์ hello โดยใช้ path เต็มรูปแบบ วิธีนี้มีข้อดีคือเรียกจากตำแหน่งใดก็ได้แต่มีข้อเสียคือ path มีความยาวทำให้เกิดความลำบากในการจดจำและสั่งรัน ส่วนการรันครั้งที่ 2 จะใช้อ้างอิงจาก path ที่ผู้ใช้อยู่ในปัจจุบันเป็นตัวตั้งต้นทำให้ path สั้นสะดวกในการพิมพ์ แต่จะมีปัญหากับผู้ใช้เริ่มต้นที่ไม่เข้าใจโครงสร้างของระบบไฟล์นี้ ตัวอย่างของปัญหาต่อไปนี้เกิดขึ้นเมื่อเรียกรันไฟล์ hello จากไดเรกทอรี่อื่น จากภาพข้างล่างการเรียกรันครั้งแรกเป็นการเรียกแบบอ้าง path เต็มรูป ส่วนแบบที่สองบ้างแบบย่อ จะเห็นได้ว่าเกิดปัญหาเพราะ path ที่ผู้ใช้อยู่ในปัจจุบันด้วยเครื่องหมายจุลภาค (.) ซึ่งก็คือ /home/pi/ แต่โปรแกรม hello จริงแล้วอยู่ที่ /home/pi/c_codes จึงเป็นการอ้างผิดที่ เมื่ออ้างผิดที่ระบบก็ไปตรวจสอบตรงที่สั่ง เมื่อไม่พบก็แจ้งว่าไม่พบไฟล์หรือไดเรกทอรี่ชื่อนี้ในตำแหน่งที่ระบุ

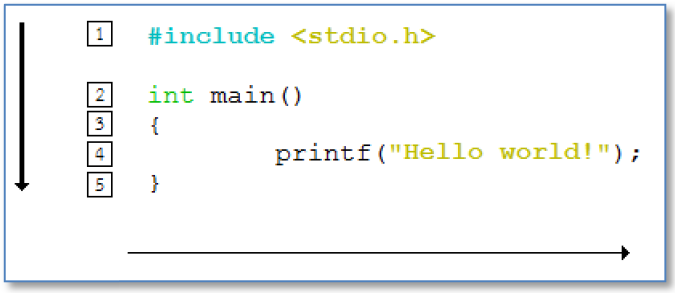


ผลรันที่ถูกต้องก็คือ



อธิบายการการทำงานโปรแกรม helloworld.c

โปรแกรมจะทำงานตามคำสั่งจากบนลงล่าง แต่ละบรรทัดจะทำจากซ้ายไปขวา



บรรทัดที่ 1 กำหนดพรีโพรเซสเซอร์	คำสั่ง #include <ชื่อไลบรารี่> เป็นการกำหนดให้เชื่อมโยงไปยังไลบรานี่ชื่อ stdio.h เพื่อเรียกใช้ฟังก์ชันในไลบรารี่ดังกล่าว ซึ่ง printf ก็เป็นฟังก์ชันในไลบรารี่นี้ด้วย ฉะนั้นถ้าเรียกใช้ printf โดยไม่ใส่บรรทัดนี้จะทำให้เกิดข้อผิดพลาดในตอนคอมไพล์
บรรทัดที่ 2 ประกาศฟังก์ชันหลัก	ฟังก์ชัน main() เป็นฟังก์ชันที่ถูกเรียกใช้อัตโนมัติมีการรันโปรแกรม จากภาพฟังก์ชันมีการส่งค่ากลับเป็นข้อมูลประเภทจำนวนเต็ม
บรรทัดที่ 3 เริ่มต้นฟังก์ชันหลัก	
บรรทัดที่ 4 เรียกใช้คำสั่ง	เรียกใช้ฟังก์ชัน printf() เพื่อแสดงข้อความที่ระบุในเครื่องหมายวงเล็บ printf(“Hello world!”) จากภาพแสดงข้อความว่า Hello World! ส่วนเครื่องหมายอัญประกาศเปิดและอัญประกาศปิด (“”) เป็นบอกให้คอมไพเลอร์รู้จุดเริ่มต้นของข้อความและจุดสิ้นสุดของข้อความ
บรรทัดที่ 5 สิ้นสุดฟังก์ชันหลัก	