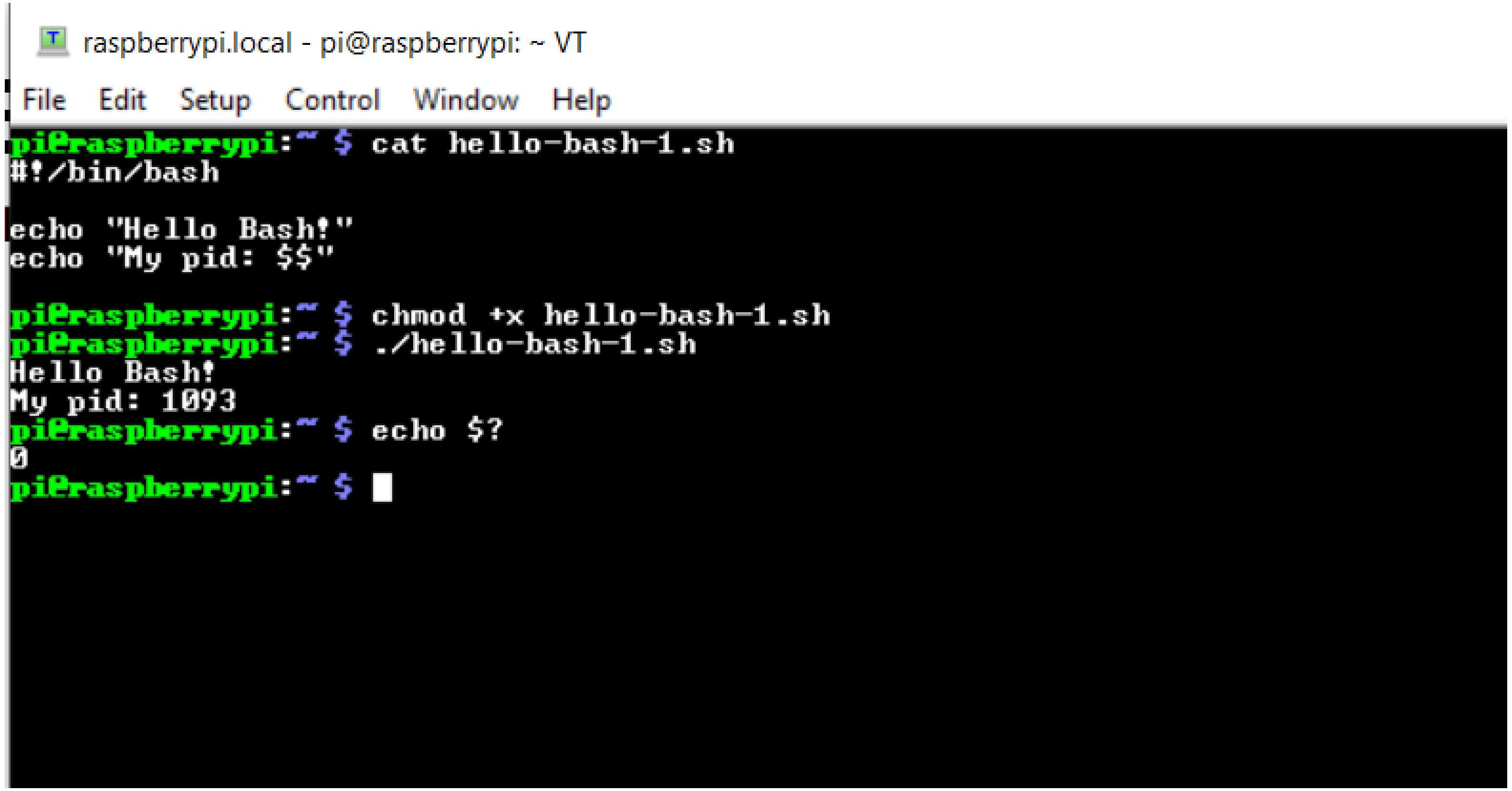
UCSD Embedded Linux Assignment 2

By

Hsuankai Chang

hsuankac@umich.edu

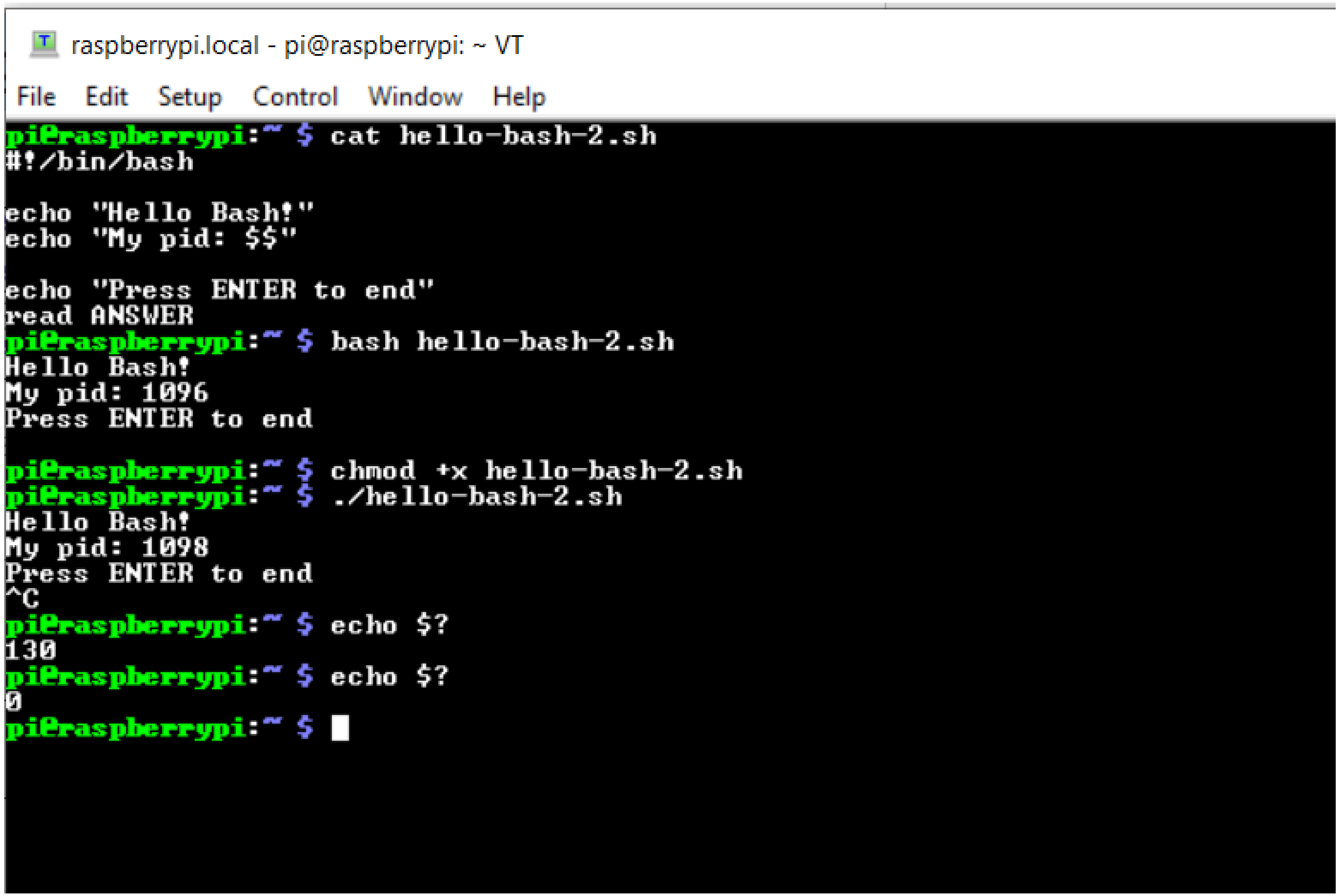
Step 1. Hello Bash - Demo 1. Foreground Program Runs to Completion



raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

```
pi@raspberrypi:~ $ cat hello-bash-1.sh
#!/bin/bash

echo "Hello Bash!"
echo "My pid: $$"

pi@raspberrypi:~ $ chmod +x hello-bash-1.sh
pi@raspberrypi:~ $ ./hello-bash-1.sh
Hello Bash!
My pid: 1093
pi@raspberrypi:~ $ echo $?
0
pi@raspberrypi:~ $
```

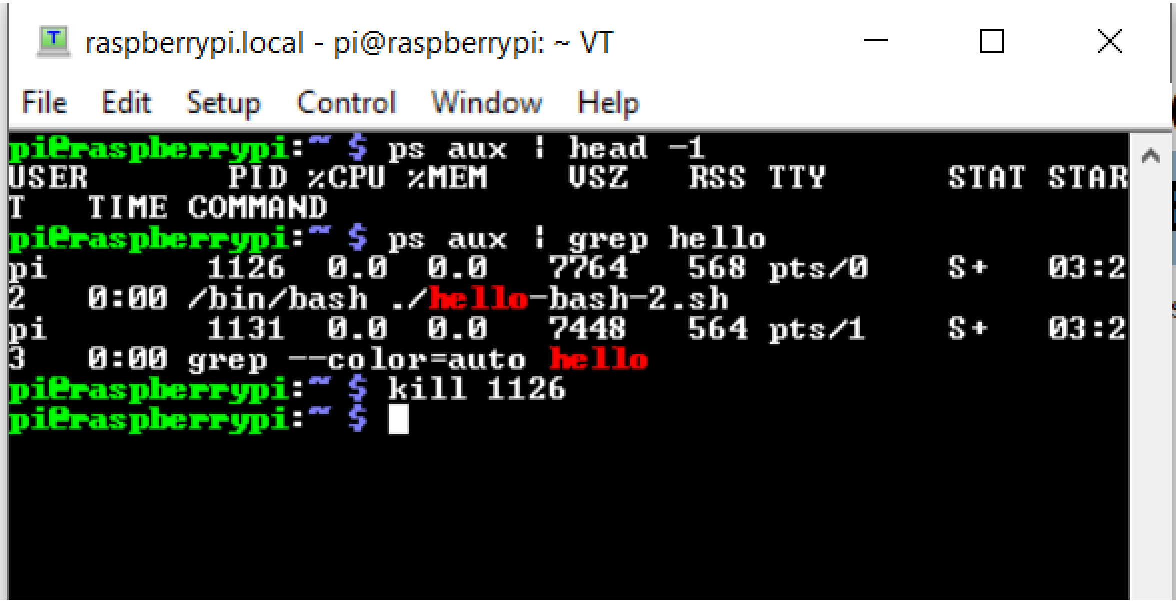
Step 2. Hello Bash - Demo 2. Foreground Program Ended by Pressing ENTER or Control+c



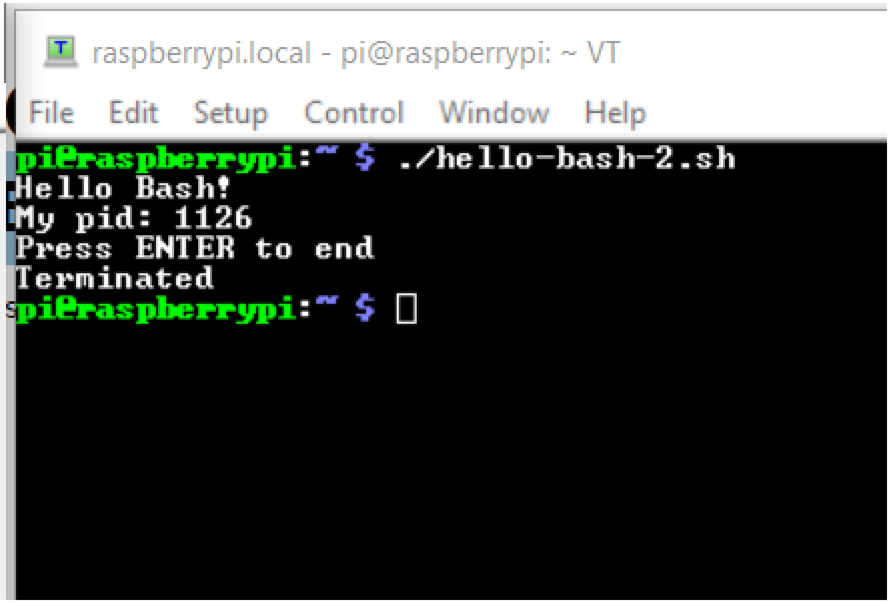
raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

```
pi@raspberrypi:~ $ cat hello-bash-2.sh
#!/bin/bash

echo "Hello Bash!"
echo "My pid: $$"

echo "Press ENTER to end"
read ANSWER
pi@raspberrypi:~ $ bash hello-bash-2.sh
Hello Bash!
My pid: 1096
Press ENTER to end

pi@raspberrypi:~ $ chmod +x hello-bash-2.sh
pi@raspberrypi:~ $ ./hello-bash-2.sh
Hello Bash!
My pid: 1098
Press ENTER to end
^C
pi@raspberrypi:~ $ echo $?
130
pi@raspberrypi:~ $ echo $?
0
pi@raspberrypi:~ $ 
```

Step 3. Hello Bash - Demo 2. Foreground Program Ended by kill

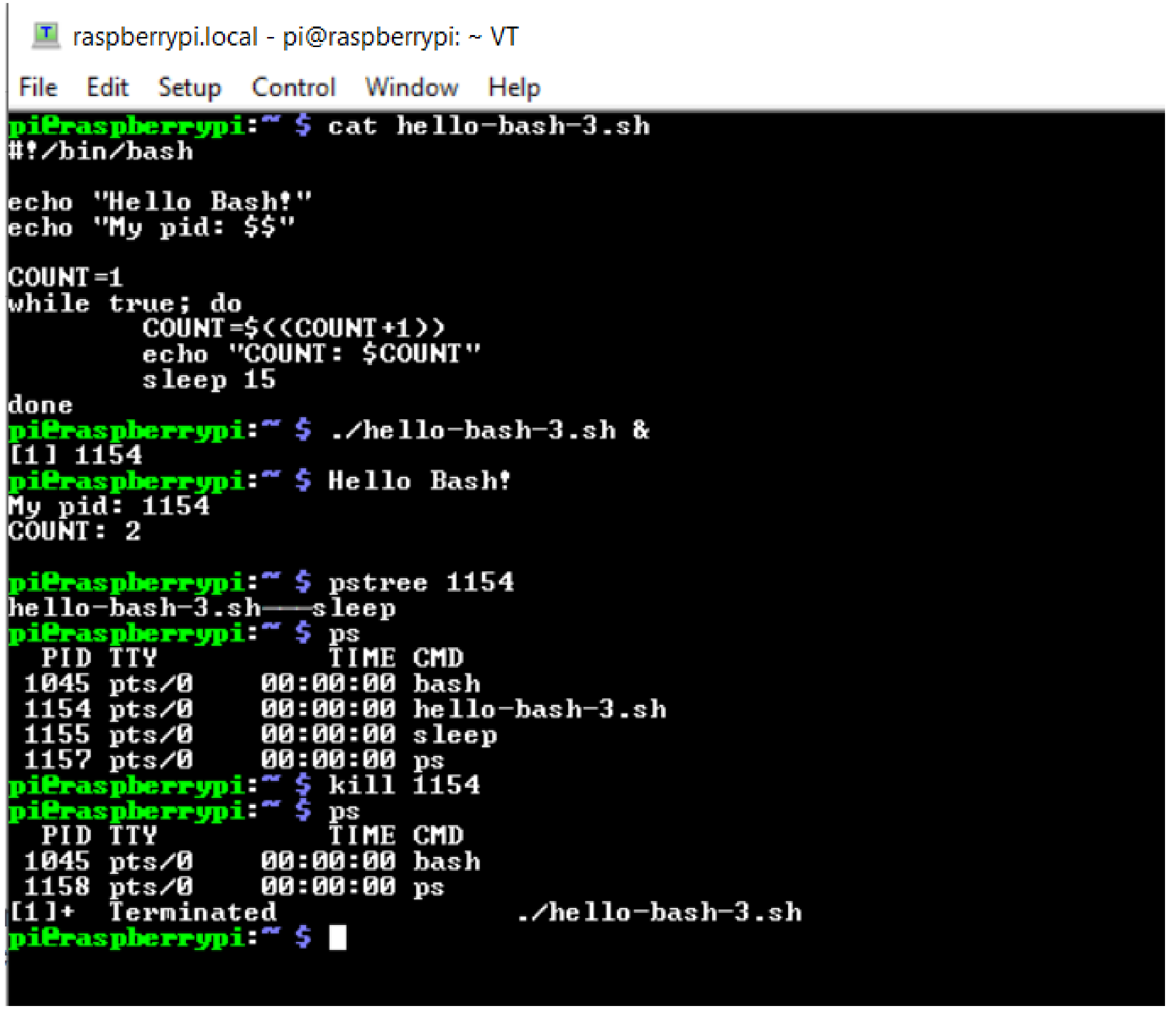


```
pi@raspberrypi:~ $ ps aux | head -1
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT STAR
T   TIME COMMAND
pi@raspberrypi:~ $ ps aux | grep hello
pi          1126  0.0  0.0   7764   568 pts/0    S+   03:2
2   0:00 /bin/bash ./hello-bash-2.sh
pi          1131  0.0  0.0   7448   564 pts/1    S+   03:2
3   0:00 grep --color=auto hello
pi@raspberrypi:~ $ kill 1126
pi@raspberrypi:~ $
```



```
pi@raspberrypi:~ $ ./hello-bash-2.sh
Hello Bash!
My pid: 1126
Press ENTER to end
Terminated
pi@raspberrypi:~ $
```

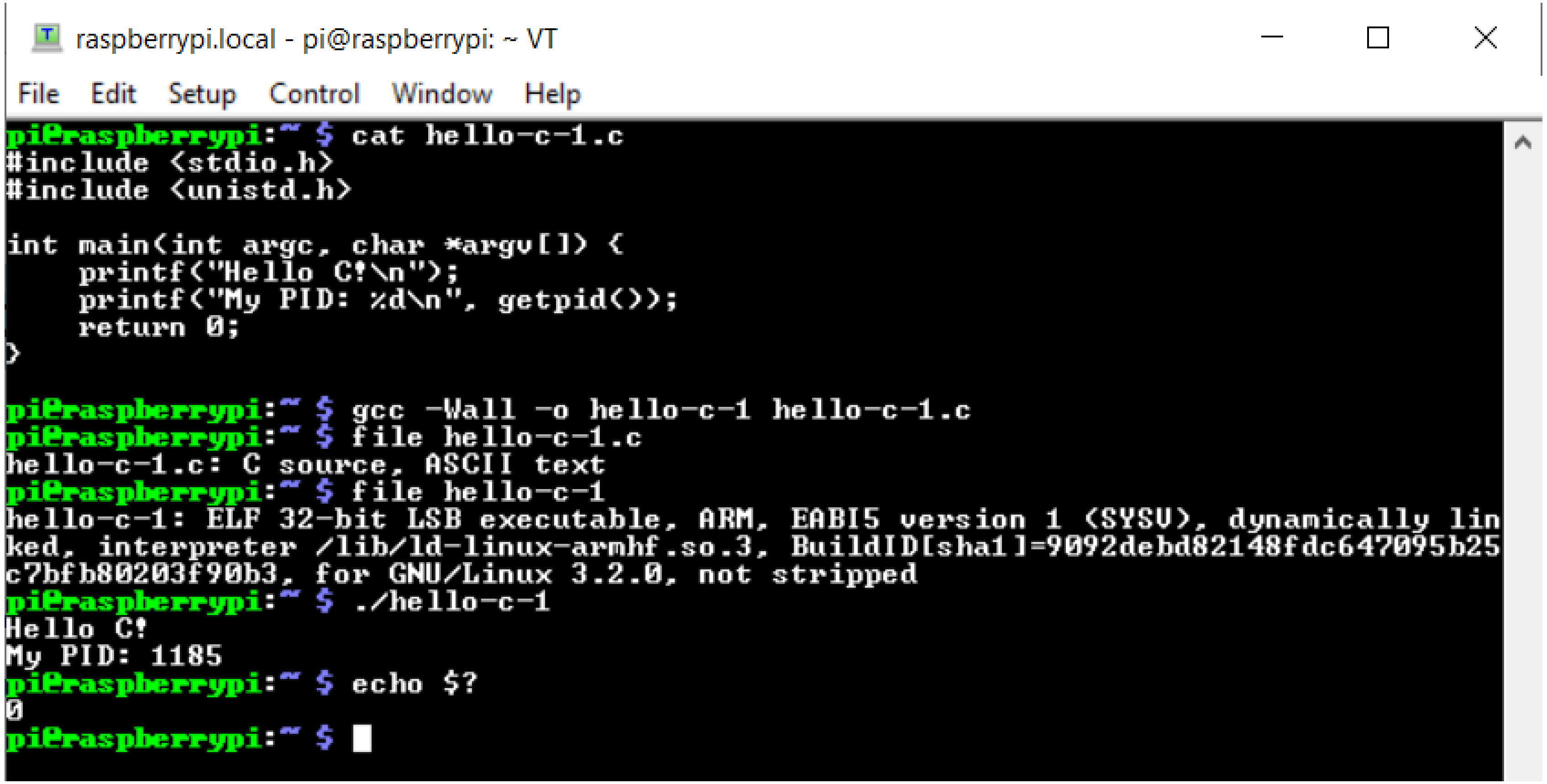
Step 4. Hello Bash - Demo 3. Run in Background



raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

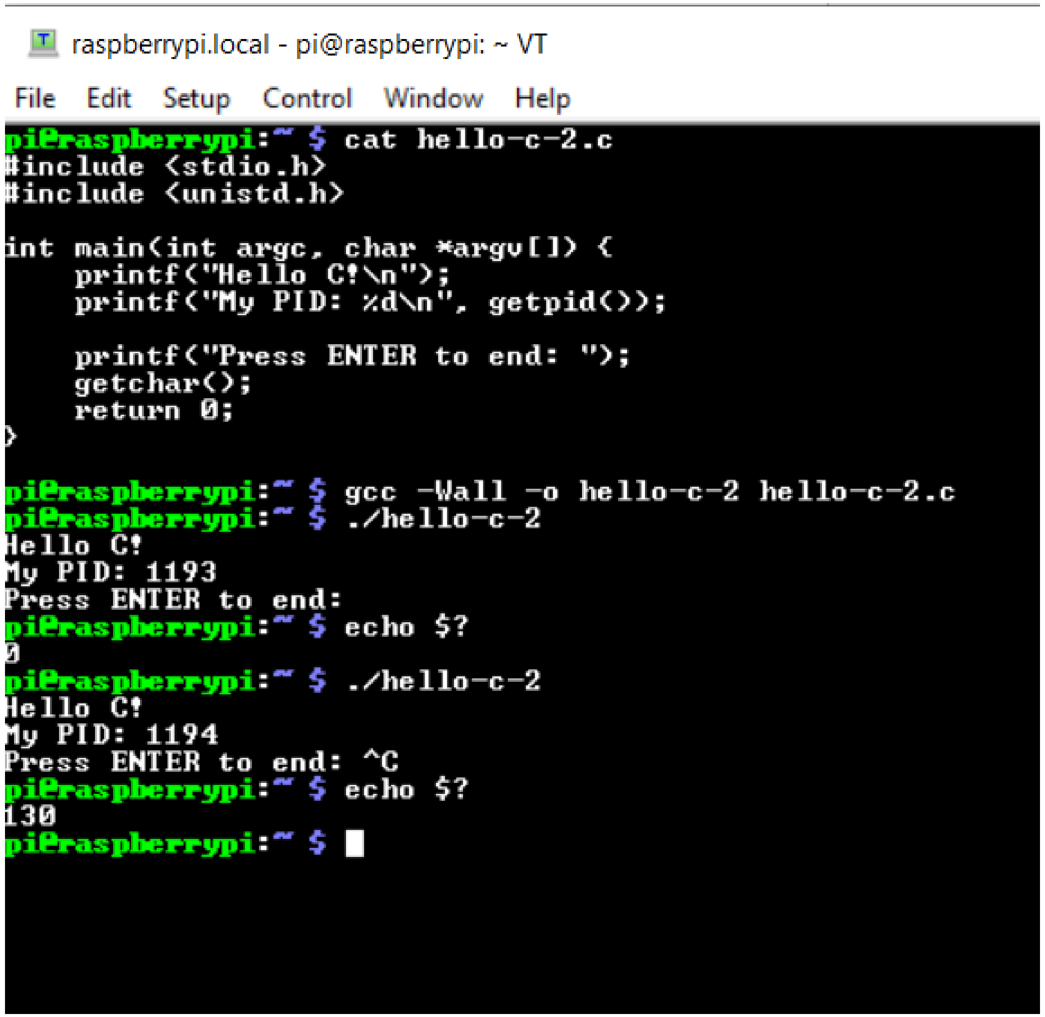
```
pi@raspberrypi:~ $ cat hello-bash-3.sh
#!/bin/bash

echo "Hello Bash!"
echo "My pid: $$"

COUNT=1
while true; do
        COUNT=$((COUNT+1))
        echo "COUNT: $COUNT"
        sleep 15
done
pi@raspberrypi:~ $ ./hello-bash-3.sh &
[1] 1154
pi@raspberrypi:~ $ Hello Bash!
My pid: 1154
COUNT: 2

pi@raspberrypi:~ $ pstree 1154
hello-bash-3.sh───sleep
pi@raspberrypi:~ $ ps
  PID TTY          TIME CMD
 1045 pts/0    00:00:00 bash
 1154 pts/0    00:00:00 hello-bash-3.sh
 1155 pts/0    00:00:00 sleep
 1157 pts/0    00:00:00 ps
pi@raspberrypi:~ $ kill 1154
pi@raspberrypi:~ $ ps
  PID TTY          TIME CMD
 1045 pts/0    00:00:00 bash
 1158 pts/0    00:00:00 ps
[1]+  Terminated              ./hello-bash-3.sh
pi@raspberrypi:~ $ 
```

Step 5. Hello C - Demo 1. Foreground Program Runs to Completion



```
pi@raspberrypi:~ $ cat hello-c-1.c
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
    printf("Hello C!\n");
    printf("My PID: %d\n", getpid());
    return 0;
}

pi@raspberrypi:~ $ gcc -Wall -o hello-c-1 hello-c-1.c
pi@raspberrypi:~ $ file hello-c-1.c
hello-c-1.c: C source, ASCII text
pi@raspberrypi:~ $ file hello-c-1
hello-c-1: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically lin
ked, interpreter /lib/ld-linux-armhf.so.3, BuildID[sha1]=9092debd82148fdc647095b25
c7bfb80203f90b3, for GNU/Linux 3.2.0, not stripped
pi@raspberrypi:~ $ ./hello-c-1
Hello C!
My PID: 1185
pi@raspberrypi:~ $ echo $?
0
pi@raspberrypi:~ $ 
```

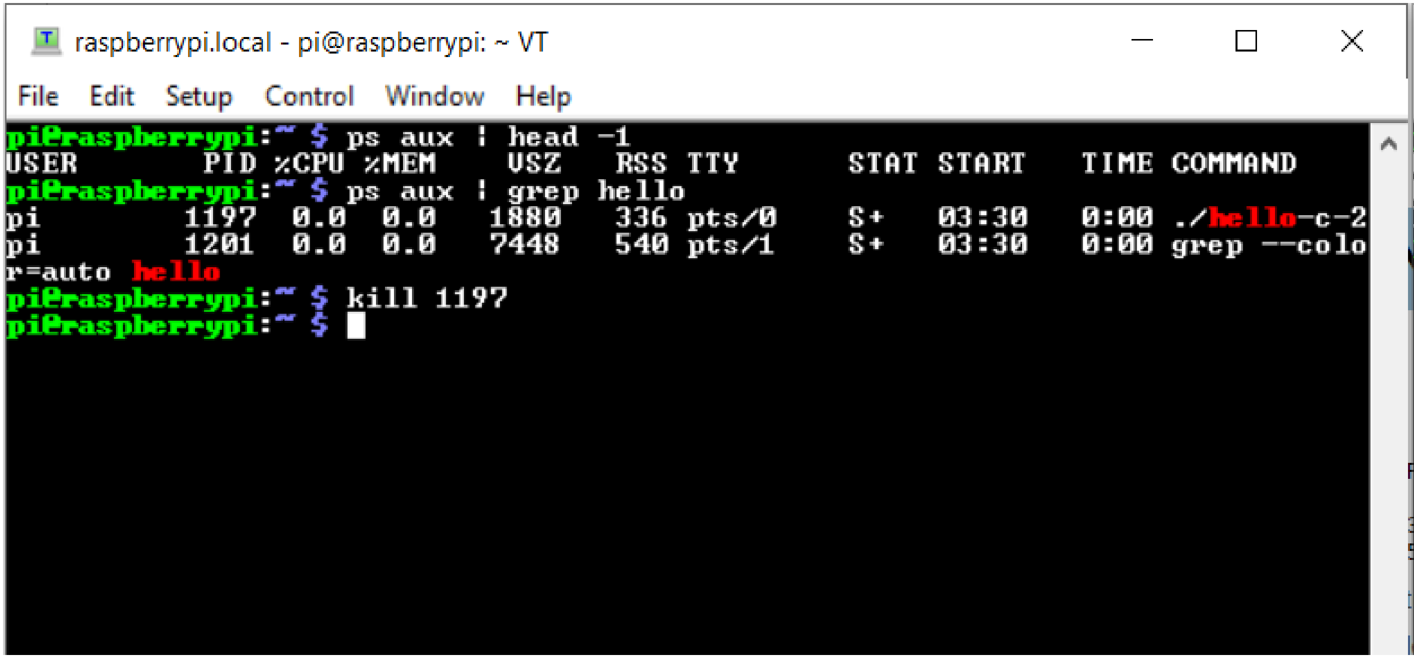
Step 6. Hello C - Demo 2. Foreground Program Ended by Pressing ENTER or Control+c



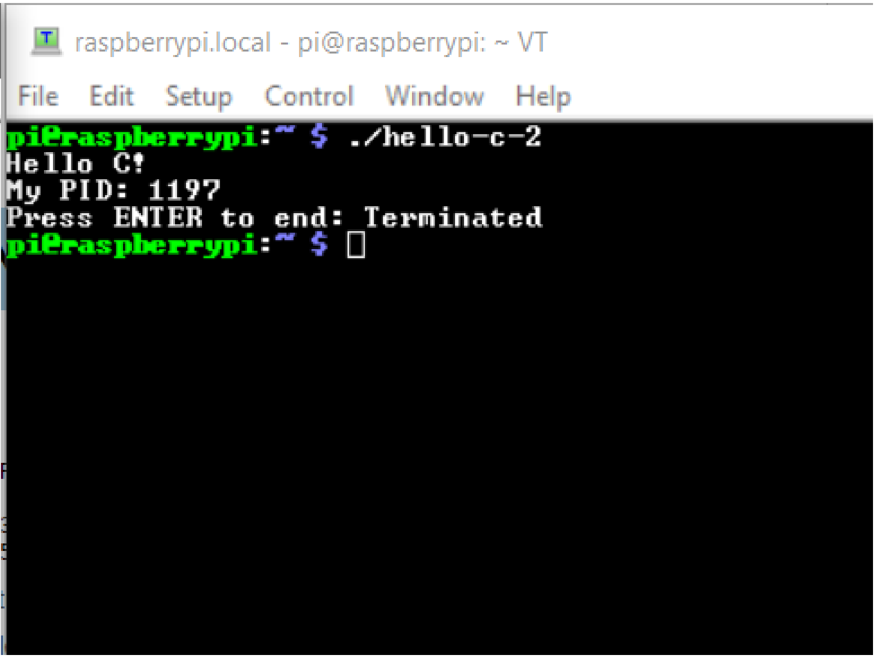
raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

```
pi@raspberrypi:~ $ cat hello-c-2.c
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
    printf("Hello C!\n");
    printf("My PID: %d\n", getpid());

    printf("Press ENTER to end: ");
    getchar();
    return 0;
}

pi@raspberrypi:~ $ gcc -Wall -o hello-c-2 hello-c-2.c
pi@raspberrypi:~ $ ./hello-c-2
Hello C!
My PID: 1193
Press ENTER to end:
pi@raspberrypi:~ $ echo $?
0
pi@raspberrypi:~ $ ./hello-c-2
Hello C!
My PID: 1194
Press ENTER to end: ^C
pi@raspberrypi:~ $ echo $?
130
pi@raspberrypi:~ $
```

Step 7. Hello C - Demo 2. Foreground Program Ended by kill

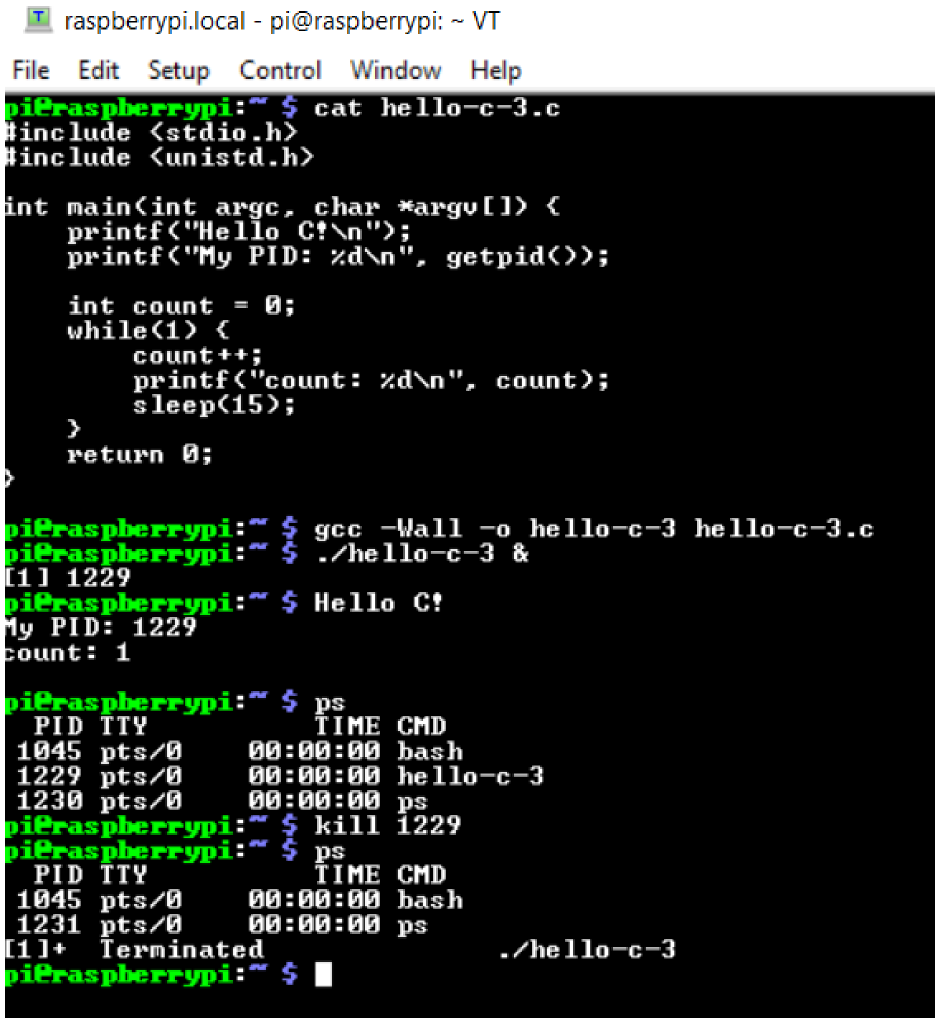

raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

```
pi@raspberrypi:~ $ ps aux | head -1
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
pi@raspberrypi:~ $ ps aux | grep hello
pi        1197  0.0  0.0   1880   336 pts/0    S+   03:30   0:00 ./hello-c-2
pi        1201  0.0  0.0   7448   540 pts/1    S+   03:30   0:00 grep --colo
r=auto hello
pi@raspberrypi:~ $ kill 1197
pi@raspberrypi:~ $
```


raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

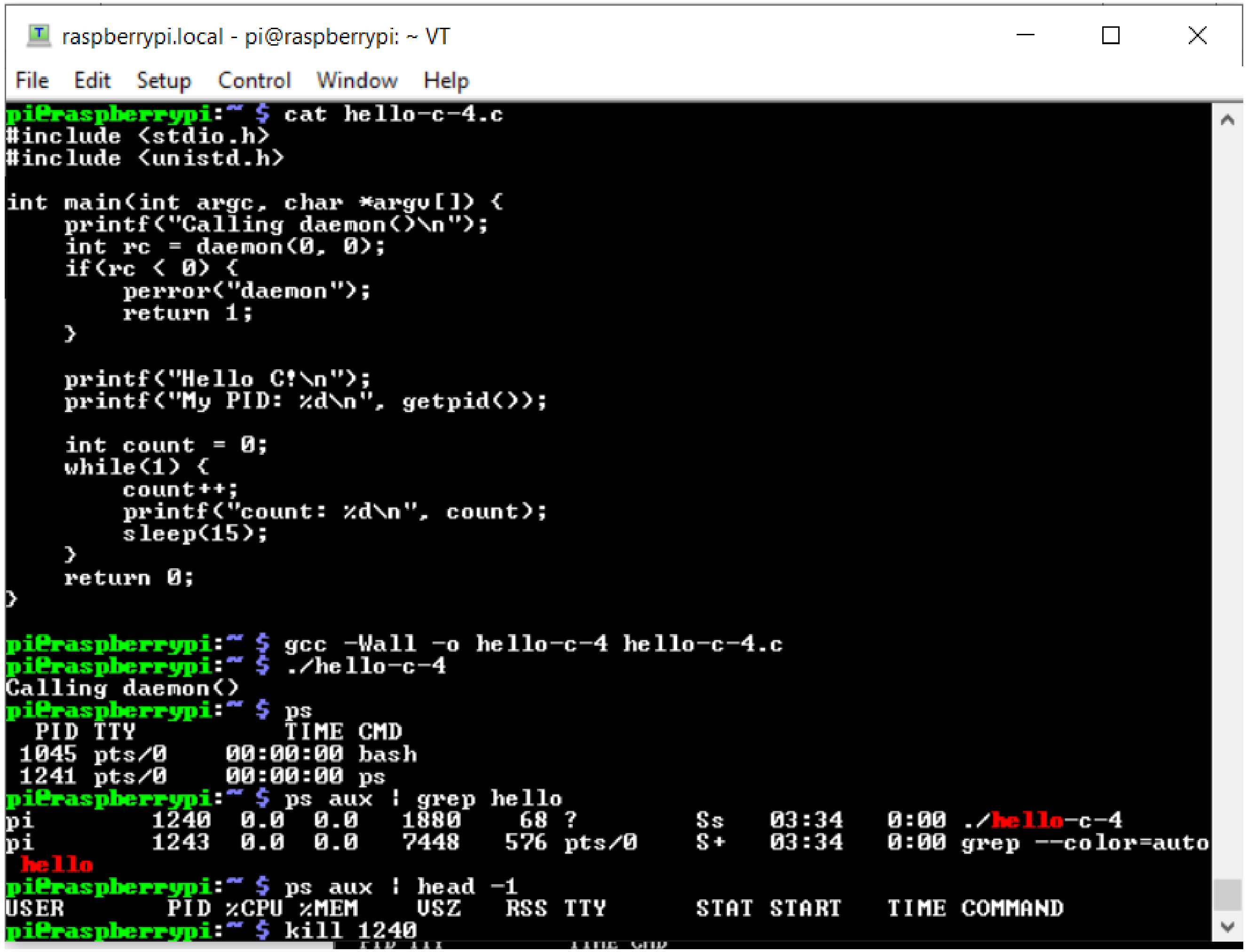
```
pi@raspberrypi:~ $ ./hello-c-2
Hello C!
My PID: 1197
Press ENTER to end: Terminated
pi@raspberrypi:~ $
```

Step 8. Hello C - Demo 3. Run in Background



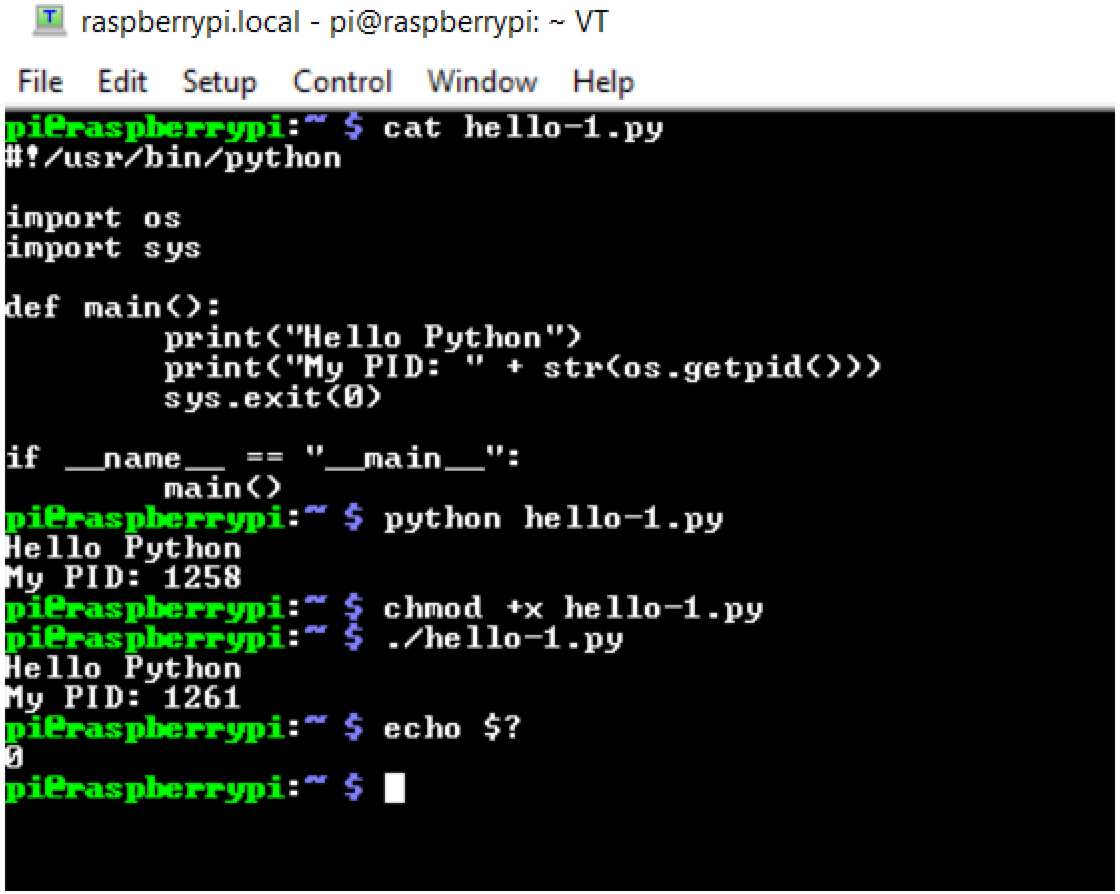
```
pi@raspberrypi:~ $ cat hello-c-3.c
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
    printf("Hello C!\n");
    printf("My PID: %d\n", getpid());

    int count = 0;
    while(1) {
        count++;
        printf("count: %d\n", count);
        sleep(15);
    }
    return 0;
}

pi@raspberrypi:~ $ gcc -Wall -o hello-c-3 hello-c-3.c
pi@raspberrypi:~ $ ./hello-c-3 &
[1] 1229
pi@raspberrypi:~ $ Hello C!
My PID: 1229
count: 1

pi@raspberrypi:~ $ ps
  PID TTY          TIME CMD
 1045 pts/0    00:00:00 bash
 1229 pts/0    00:00:00 hello-c-3
 1230 pts/0    00:00:00 ps
pi@raspberrypi:~ $ kill 1229
pi@raspberrypi:~ $ ps
  PID TTY          TIME CMD
 1045 pts/0    00:00:00 bash
 1231 pts/0    00:00:00 ps
[1]+  Terminated              ./hello-c-3
pi@raspberrypi:~ $ █
```

Step 9. Hello C - Demo 4. Run as Daemon


raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

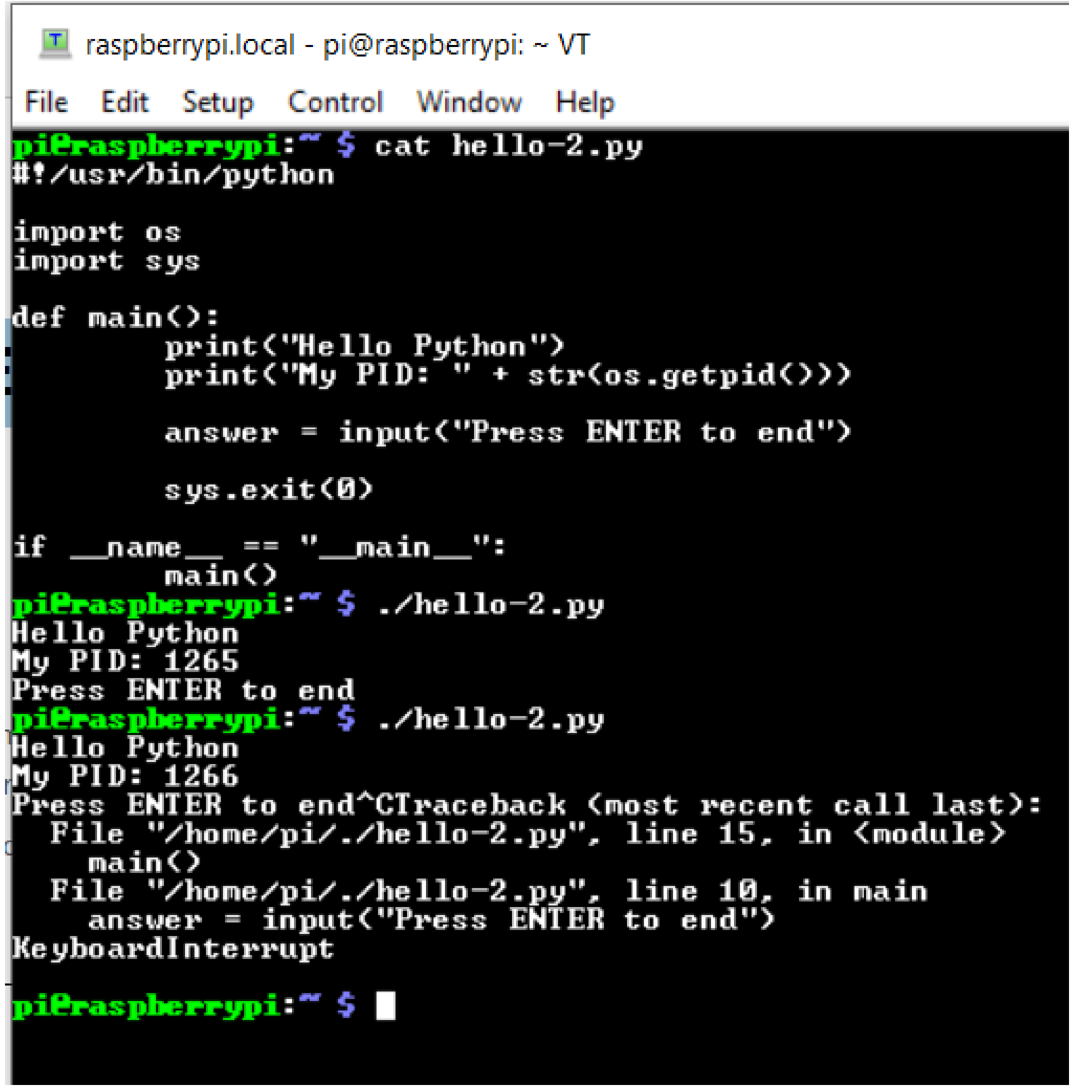
```
pi@raspberrypi:~ $ cat hello-c-4.c
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
    printf("Calling daemon()\n");
    int rc = daemon(0, 0);
    if(rc < 0) {
        perror("daemon");
        return 1;
    }

    printf("Hello C!\n");
    printf("My PID: %d\n", getpid());

    int count = 0;
    while(1) {
        count++;
        printf("count: %d\n", count);
        sleep(15);
    }
    return 0;
}

pi@raspberrypi:~ $ gcc -Wall -o hello-c-4 hello-c-4.c
pi@raspberrypi:~ $ ./hello-c-4
Calling daemon()
pi@raspberrypi:~ $ ps
  PID TTY          TIME CMD
 1045 pts/0    00:00:00 bash
 1241 pts/0    00:00:00 ps
pi@raspberrypi:~ $ ps aux | grep hello
pi        1240  0.0  0.0   1880    68 ?        Ss   03:34   0:00 ./hello-c-4
pi        1243  0.0  0.0   7448   576 pts/0    S+   03:34   0:00 grep --color=auto
 hello
pi@raspberrypi:~ $ ps aux | head -1
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
pi@raspberrypi:~ $ kill 1240
```

Step 10. Hello Python - Demo 1. Foreground program runs to completion


raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

```
pi@raspberrypi:~ $ cat hello-1.py
#!/usr/bin/python

import os
import sys

def main():
        print("Hello Python")
        print("My PID: " + str(os.getpid()))
        sys.exit(0)

if __name__ == "__main__":
        main()
pi@raspberrypi:~ $ python hello-1.py
Hello Python
My PID: 1258
pi@raspberrypi:~ $ chmod +x hello-1.py
pi@raspberrypi:~ $ ./hello-1.py
Hello Python
My PID: 1261
pi@raspberrypi:~ $ echo $?
0
pi@raspberrypi:~ $ █
```

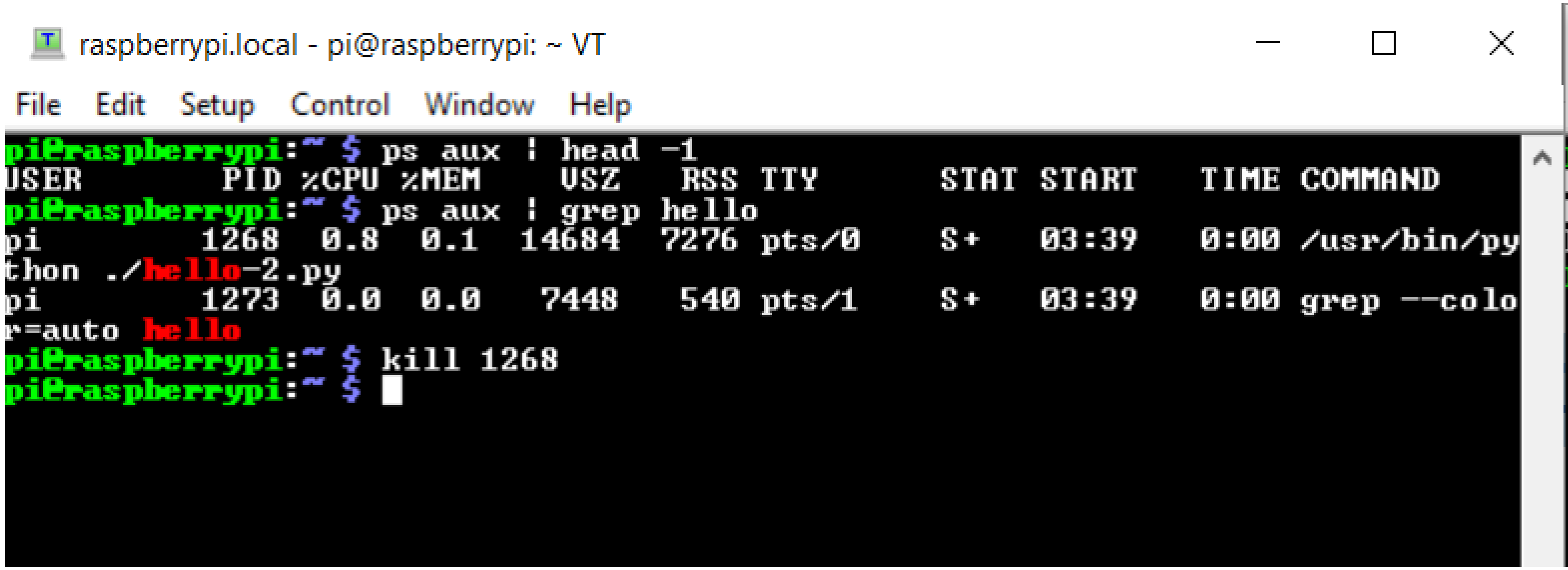
Step 11. Hello Python - Demo 2. Foreground Program Ended by Pressing ENTER or Control+c



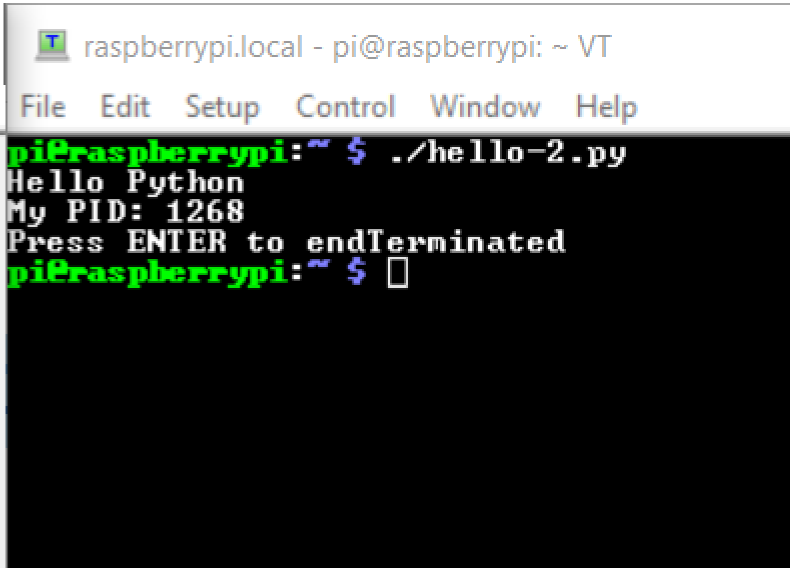
raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

```
pi@raspberrypi:~ $ cat hello-2.py
#!/usr/bin/python

import os
import sys

def main():
        print("Hello Python")
        print("My PID: " + str(os.getpid()))

        answer = input("Press ENTER to end")

        sys.exit(0)

if __name__ == "__main__":
        main()
pi@raspberrypi:~ $ ./hello-2.py
Hello Python
My PID: 1265
Press ENTER to end
pi@raspberrypi:~ $ ./hello-2.py
Hello Python
My PID: 1266
Press ENTER to end^CTraceback (most recent call last):
  File "/home/pi/./hello-2.py", line 15, in <module>
    main()
  File "/home/pi/./hello-2.py", line 10, in main
    answer = input("Press ENTER to end")
KeyboardInterrupt

pi@raspberrypi:~ $
```

Step 12. Hello Python - Demo 2. Foreground Program Ended by kill

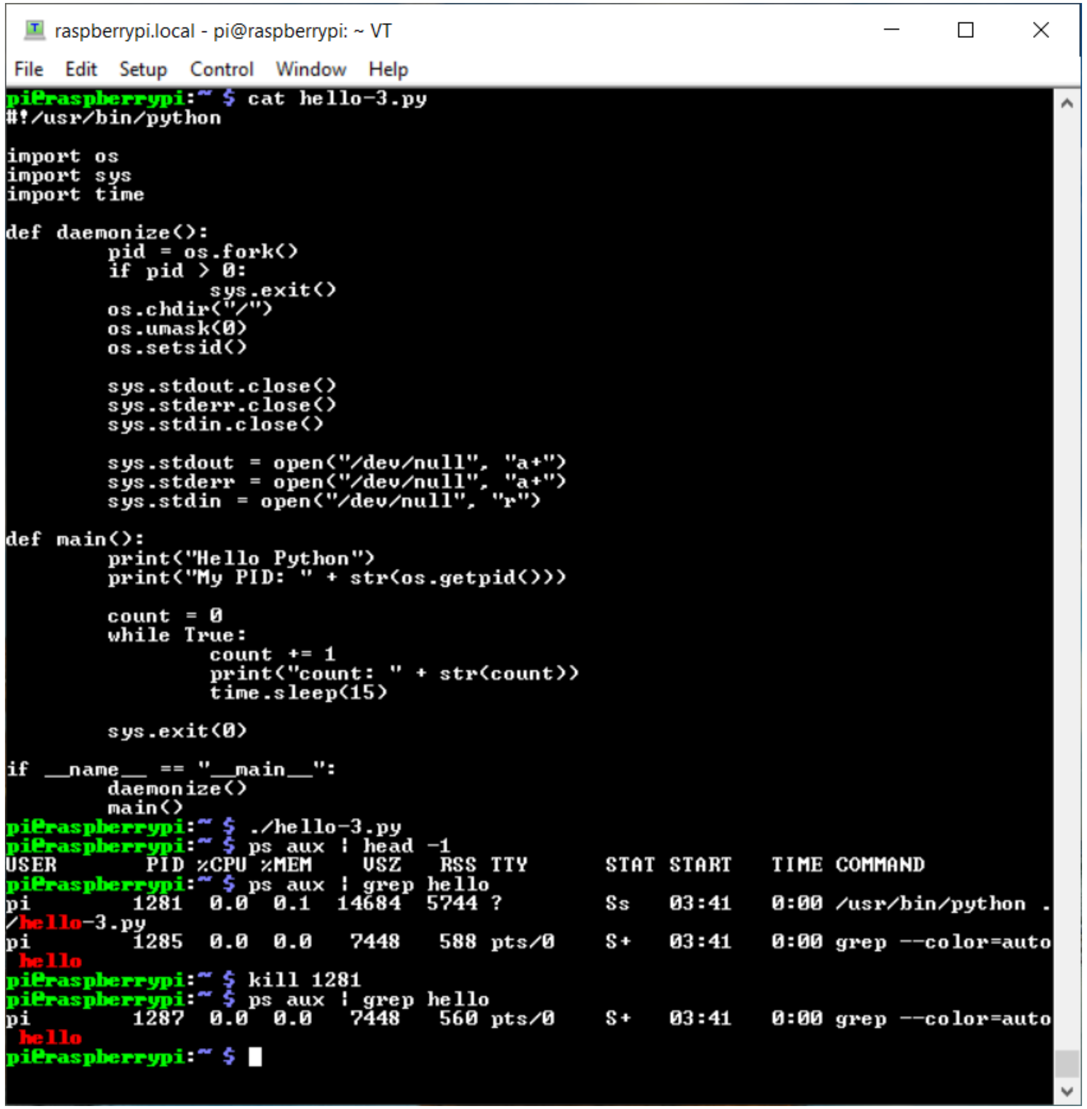

raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

```
pi@raspberrypi:~ $ ps aux | head -1
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
pi@raspberrypi:~ $ ps aux | grep hello
pi        1268  0.8  0.1  14684  7276 pts/0    S+   03:39   0:00 /usr/bin/py
thon ./hello-2.py
pi        1273  0.0  0.0   7448   540 pts/1    S+   03:39   0:00 grep --colo
r=auto hello
pi@raspberrypi:~ $ kill 1268
pi@raspberrypi:~ $
```


raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

```
pi@raspberrypi:~ $ ./hello-2.py
Hello Python
My PID: 1268
Press ENTER to endTerminated
pi@raspberrypi:~ $
```

Step 13. Hello Python - Demo 3. Run Python Program as Daemon


raspberrypi.local - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

```
pi@raspberrypi:~ $ cat hello-3.py
#!/usr/bin/python

import os
import sys
import time

def daemonize():
        pid = os.fork()
        if pid > 0:
                sys.exit()
        os.chdir("/")
        os.umask(0)
        os.setsid()

        sys.stdout.close()
        sys.stderr.close()
        sys.stdin.close()

        sys.stdout = open("/dev/null", "a+")
        sys.stderr = open("/dev/null", "a+")
        sys.stdin = open("/dev/null", "r")

def main():
        print("Hello Python")
        print("My PID: " + str(os.getpid()))

        count = 0
        while True:
                count += 1
                print("count: " + str(count))
                time.sleep(15)

        sys.exit(0)

if __name__ == "__main__":
        daemonize()
        main()
pi@raspberrypi:~ $ ./hello-3.py
pi@raspberrypi:~ $ ps aux | head -1
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
pi@raspberrypi:~ $ ps aux | grep hello
pi        1281  0.0  0.1  14684  5744 ?        Ss   03:41   0:00 /usr/bin/python .
/hello-3.py
pi        1285  0.0  0.0   7448   588 pts/0    S+   03:41   0:00 grep --color=auto
 hello
pi@raspberrypi:~ $ kill 1281
pi@raspberrypi:~ $ ps aux | grep hello
pi        1287  0.0  0.0   7448   560 pts/0    S+   03:41   0:00 grep --color=auto
 hello
pi@raspberrypi:~ $
```