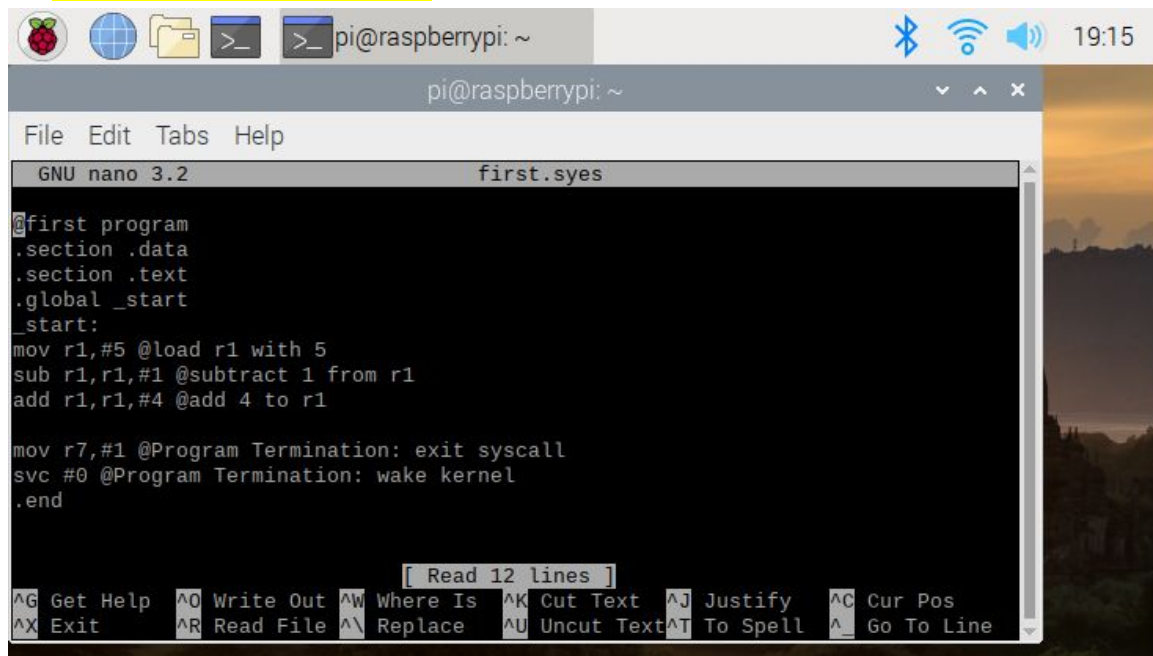


Beginning of Supriya Arun

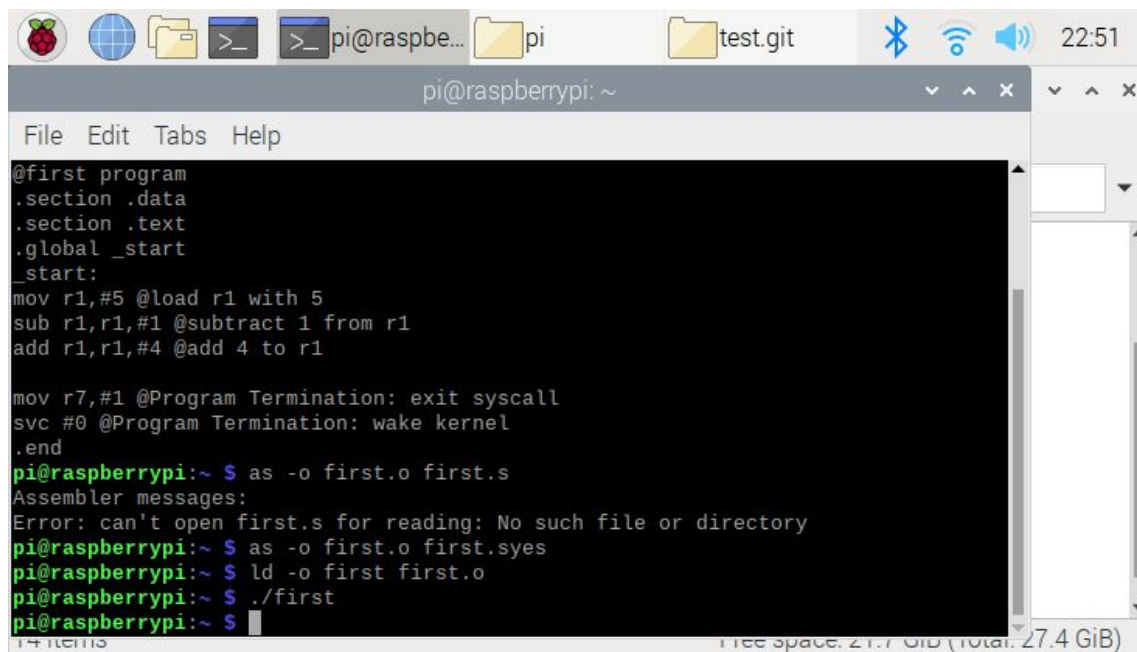


```
GNU nano 3.2 first.syes
@first program
.section .data
.section .text
.global _start
_start:
mov r1,#5 @load r1 with 5
sub r1,r1,#1 @subtract 1 from r1
add r1,r1,#4 @add 4 to r1

mov r7,#1 @Program Termination: exit syscall
svc #0 @Program Termination: wake kernel
.end

[ Read 12 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Source code for first.s (first program) in the nano editor

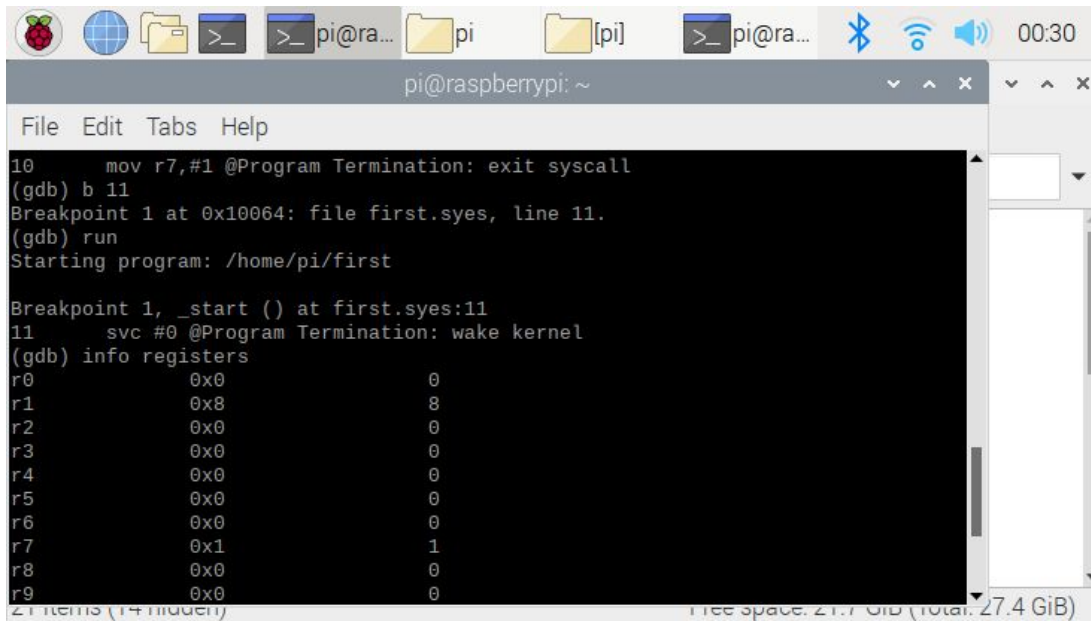


```
pi@raspberrypi: ~
File Edit Tabs Help
@first program
.section .data
.section .text
.global _start
_start:
mov r1,#5 @load r1 with 5
sub r1,r1,#1 @subtract 1 from r1
add r1,r1,#4 @add 4 to r1

mov r7,#1 @Program Termination: exit syscall
svc #0 @Program Termination: wake kernel
.end

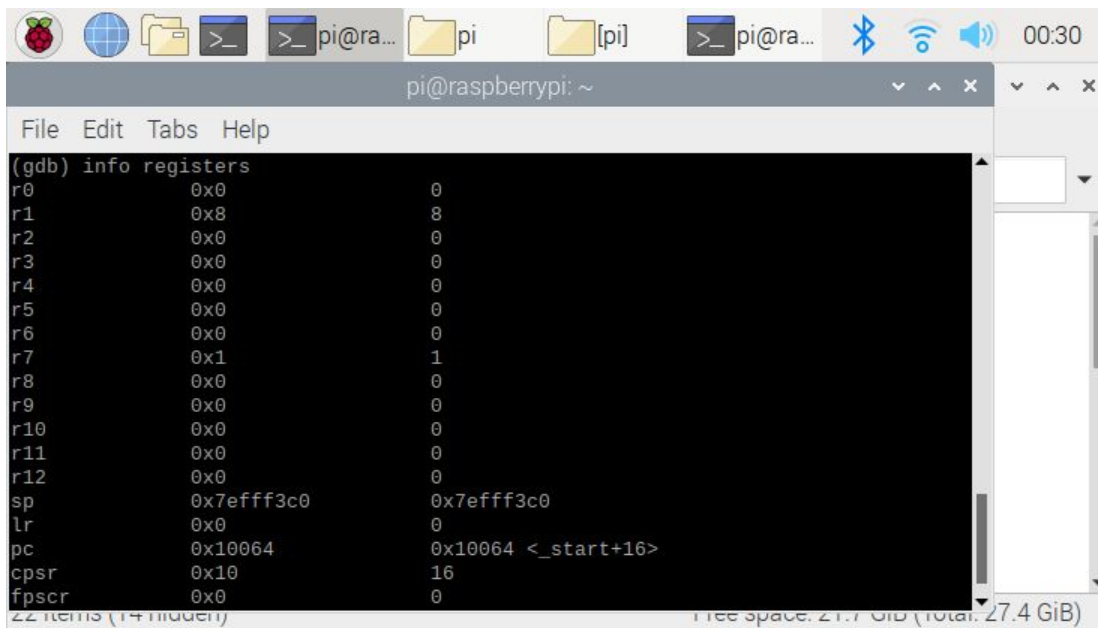
pi@raspberrypi:~ $ as -o first.o first.s
Assembler messages:
Error: can't open first.s for reading: No such file or directory
pi@raspberrypi:~ $ as -o first.o first.syes
pi@raspberrypi:~ $ ld -o first first.o
pi@raspberrypi:~ $ ./first
pi@raspberrypi:~ $
```

There was no output after entering `./first` because the code in `first.s` only manipulates the numbers in the registers and does not print or return any values. Therefore, there is no output.



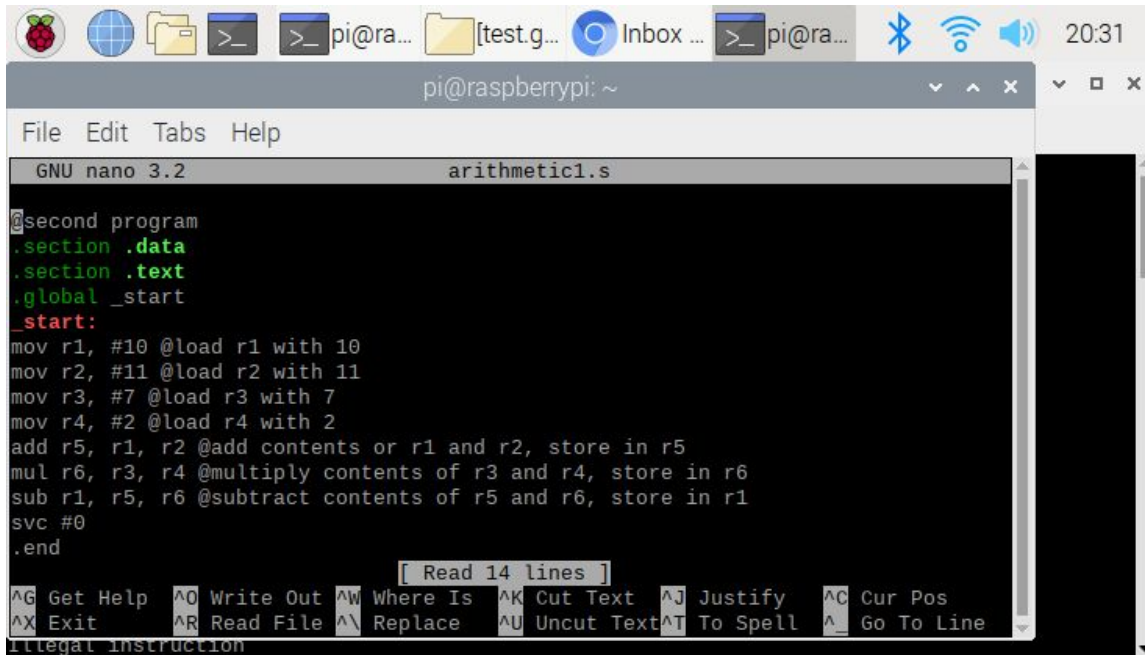
```
pi@raspberrypi: ~  
File Edit Tabs Help  
10      mov r7,#1 @Program Termination: exit syscall  
(gdb) b 11  
Breakpoint 1 at 0x10064: file first.syes, line 11.  
(gdb) run  
Starting program: /home/pi/first  
  
Breakpoint 1, _start () at first.syes:11  
11      svc #0 @Program Termination: wake kernel  
(gdb) info registers  
r0          0x0          0  
r1          0x8          8  
r2          0x0          0  
r3          0x0          0  
r4          0x0          0  
r5          0x0          0  
r6          0x0          0  
r7          0x1          1  
r8          0x0          0  
r9          0x0          0  
21 items (14 hidden) Free space: 21.7 GiB (total: 27.4 GiB)
```

In order to view the registers, it is important to stop the program execution which is done by setting a breakpoint. So, `b 11` will stop the program execution before line 11.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
(gdb) info registers  
r0          0x0          0  
r1          0x8          8  
r2          0x0          0  
r3          0x0          0  
r4          0x0          0  
r5          0x0          0  
r6          0x0          0  
r7          0x1          1  
r8          0x0          0  
r9          0x0          0  
r10         0x0          0  
r11         0x0          0  
r12         0x0          0  
sp          0x7efff3c0    0x7efff3c0  
lr          0x0          0  
pc          0x10064      0x10064 <_start+16>  
cpsr        0x10        16  
fpscr       0x0          0  
22 items (14 hidden) Free space: 21.7 GiB (total: 27.4 GiB)
```

Since the code was dealing with only registers `r1` and `r7`, it is clearly shown that all registers are empty except `r1` and `r7`. Based on the source code(load 5 into `r1`, subtract 1 from `r1`, add 4 to `r1`=8 in `r1`; load 1 into `r7`=1 in `r7`) , we can conclude that the output is valid.

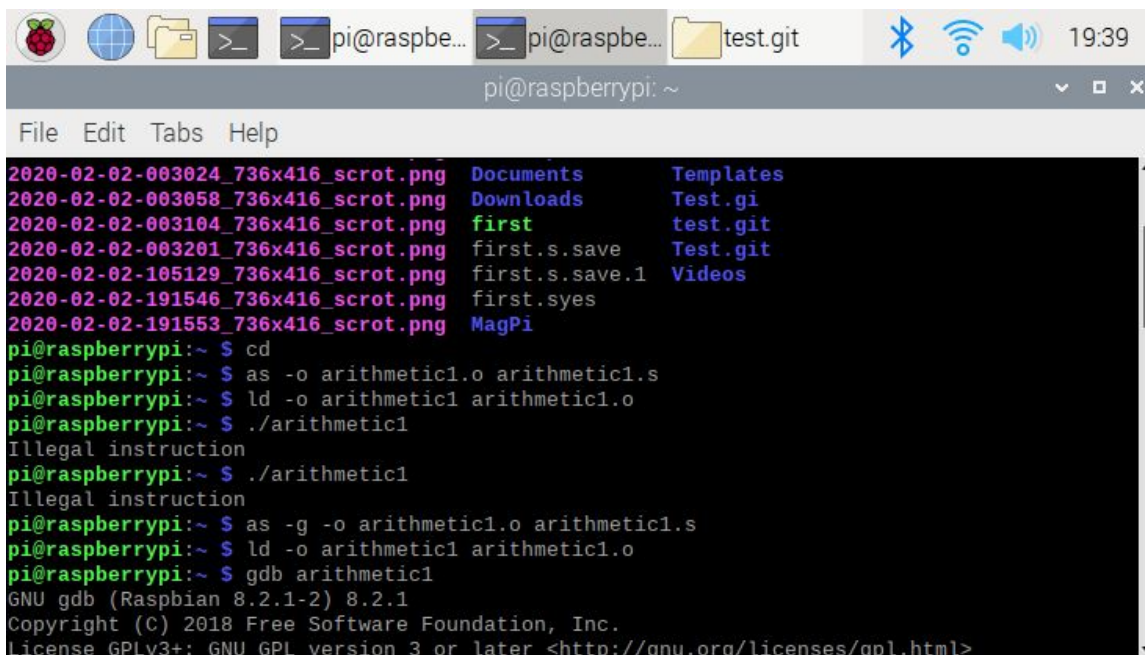


```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 3.2 arithmetic1.s

@second program
.section .data
.section .text
.global _start
_start:
mov r1, #10 @load r1 with 10
mov r2, #11 @load r2 with 11
mov r3, #7 @load r3 with 7
mov r4, #2 @load r4 with 2
add r5, r1, r2 @add contents of r1 and r2, store in r5
mul r6, r3, r4 @multiply contents of r3 and r4, store in r6
sub r1, r5, r6 @subtract contents of r5 and r6, store in r1
svc #0
.end

[ Read 14 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
Illegal instruction
```

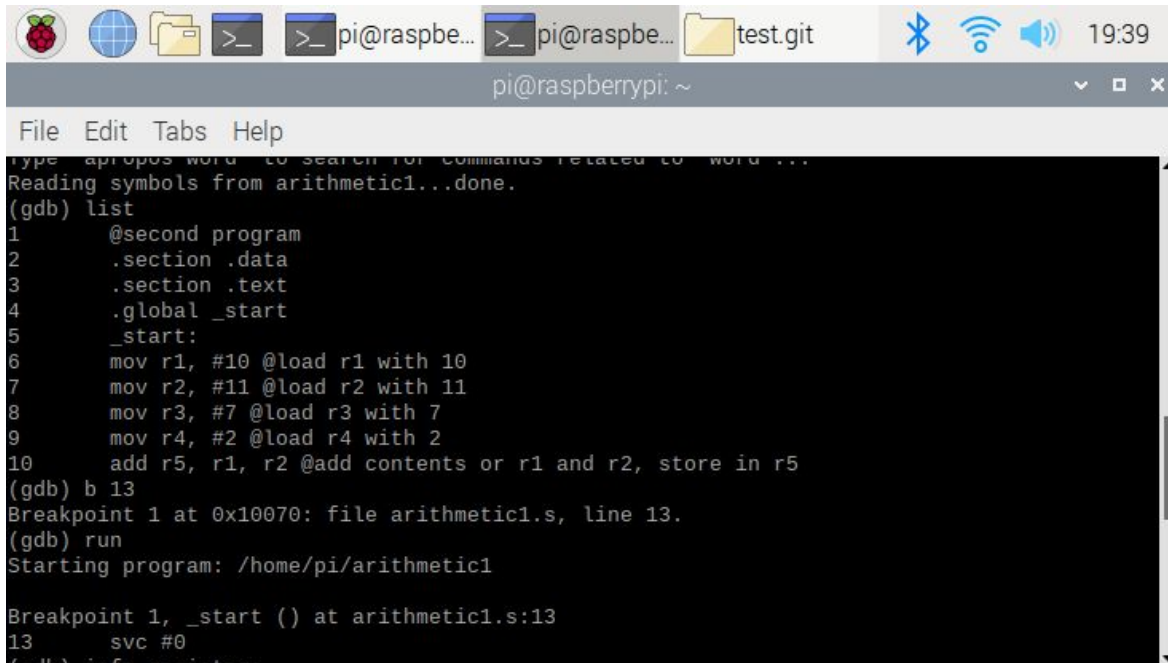
Source code for arithmetic1 (second program) in the nano editor



```
pi@raspberrypi: ~
File Edit Tabs Help
2020-02-02-003024_736x416_screenshot.png Documents Templates
2020-02-02-003058_736x416_screenshot.png Downloads Test.gi
2020-02-02-003104_736x416_screenshot.png first test.git
2020-02-02-003201_736x416_screenshot.png first.s.save Test.git
2020-02-02-105129_736x416_screenshot.png first.s.save.1 Videos
2020-02-02-191546_736x416_screenshot.png first.syes
2020-02-02-191553_736x416_screenshot.png MagPi

pi@raspberrypi:~ $ cd
pi@raspberrypi:~ $ as -o arithmetic1.o arithmetic1.s
pi@raspberrypi:~ $ ld -o arithmetic1 arithmetic1.o
pi@raspberrypi:~ $ ./arithmetic1
Illegal instruction
pi@raspberrypi:~ $ ./arithmetic1
Illegal instruction
pi@raspberrypi:~ $ as -g -o arithmetic1.o arithmetic1.s
pi@raspberrypi:~ $ ld -o arithmetic1 arithmetic1.o
pi@raspberrypi:~ $ gdb arithmetic1
GNU gdb (Raspbian 8.2.1-2) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
```

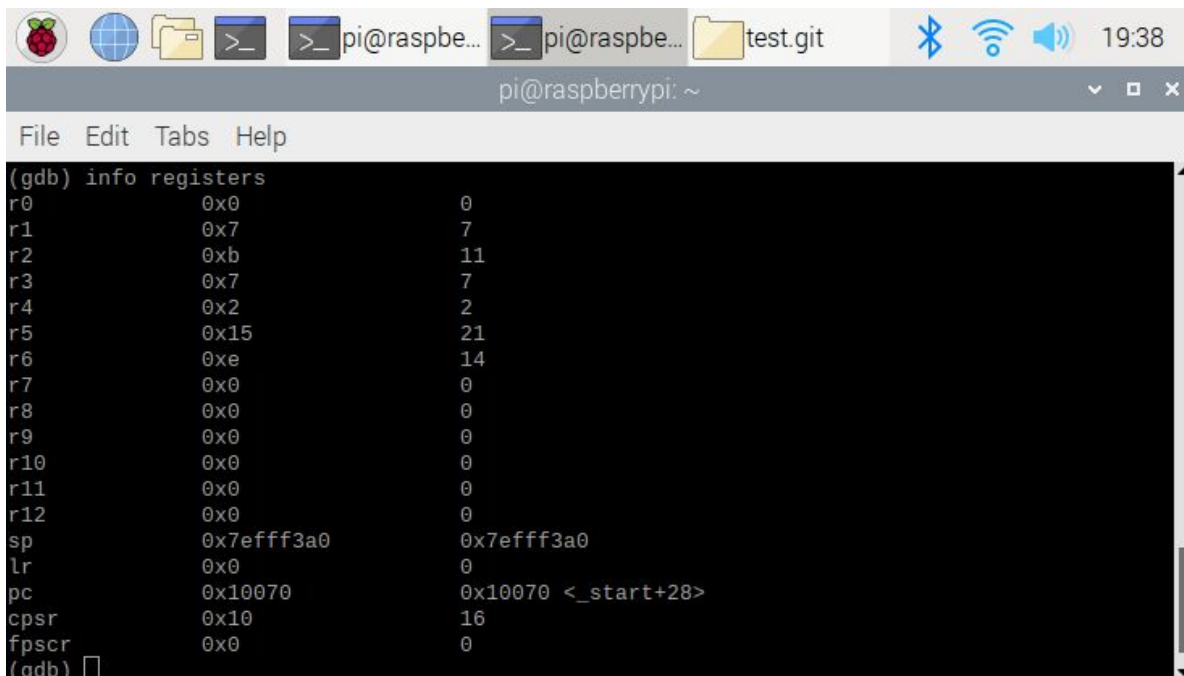
Command `./arithmetic1` does not execute anything because the code in `arithmetic1.s` just manipulates the numbers in the registers and does not print or return any values. Therefore, there is no output.



```
File Edit Tabs Help
type apropos word to search for commands related to word ...
Reading symbols from arithmetic1...done.
(gdb) list
1      @second program
2      .section .data
3      .section .text
4      .global _start
5      _start:
6      mov r1, #10 @load r1 with 10
7      mov r2, #11 @load r2 with 11
8      mov r3, #7 @load r3 with 7
9      mov r4, #2 @load r4 with 2
10     add r5, r1, r2 @add contents of r1 and r2, store in r5
(gdb) b 13
Breakpoint 1 at 0x10070: file arithmetic1.s, line 13.
(gdb) run
Starting program: /home/pi/arithmetic1

Breakpoint 1, _start () at arithmetic1.s:13
13     svc #0
(gdb) info registers
```

Command b 13 was set as a breakpoint so that the program does not terminate after the arithmetic operations. And by this command, the registers can be examined while the program is running.



```
File Edit Tabs Help
(gdb) info registers
r0          0x0          0
r1          0x7          7
r2          0xb          11
r3          0x7          7
r4          0x2          2
r5          0x15         21
r6          0xe          14
r7          0x0          0
r8          0x0          0
r9          0x0          0
r10         0x0          0
r11         0x0          0
r12         0x0          0
sp          0x7efff3a0    0x7efff3a0
lr          0x0          0
pc          0x10070      0x10070 <_start+28>
cpsr       0x10         16
fpscr      0x0          0
(gdb)
```

Registers of arithmetic1. Based on the source code, the output is valid because r2, r3, and r4 have the values of B, C, and D respectively and r1 has the final value of A (7), rather than the initial value of A (10). r5 shows the result of A+B (10+11) which is 21, r6 shows the result of C*D (7*2) which is 14 and r1 has the result of r5-r6 (21-14) which is 7.

Ending of Supriya Arun