

Michael Knight

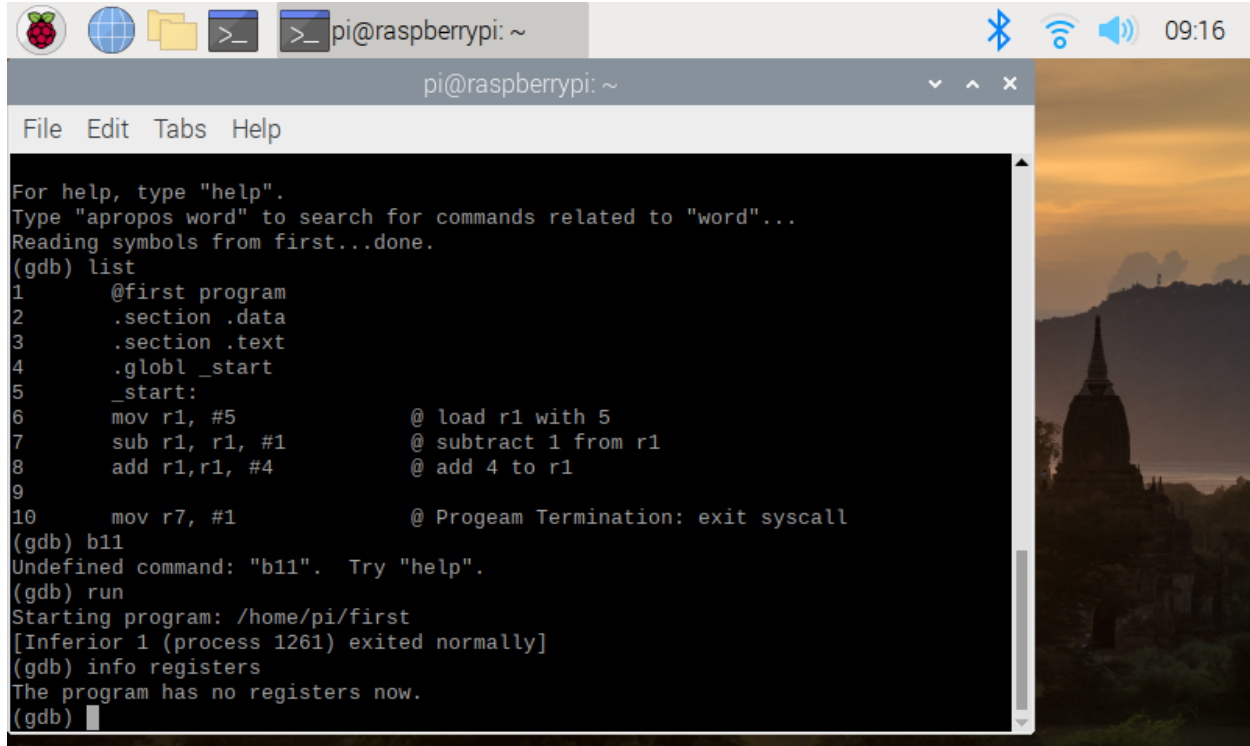
February 10 2020

Gitter-Geeks

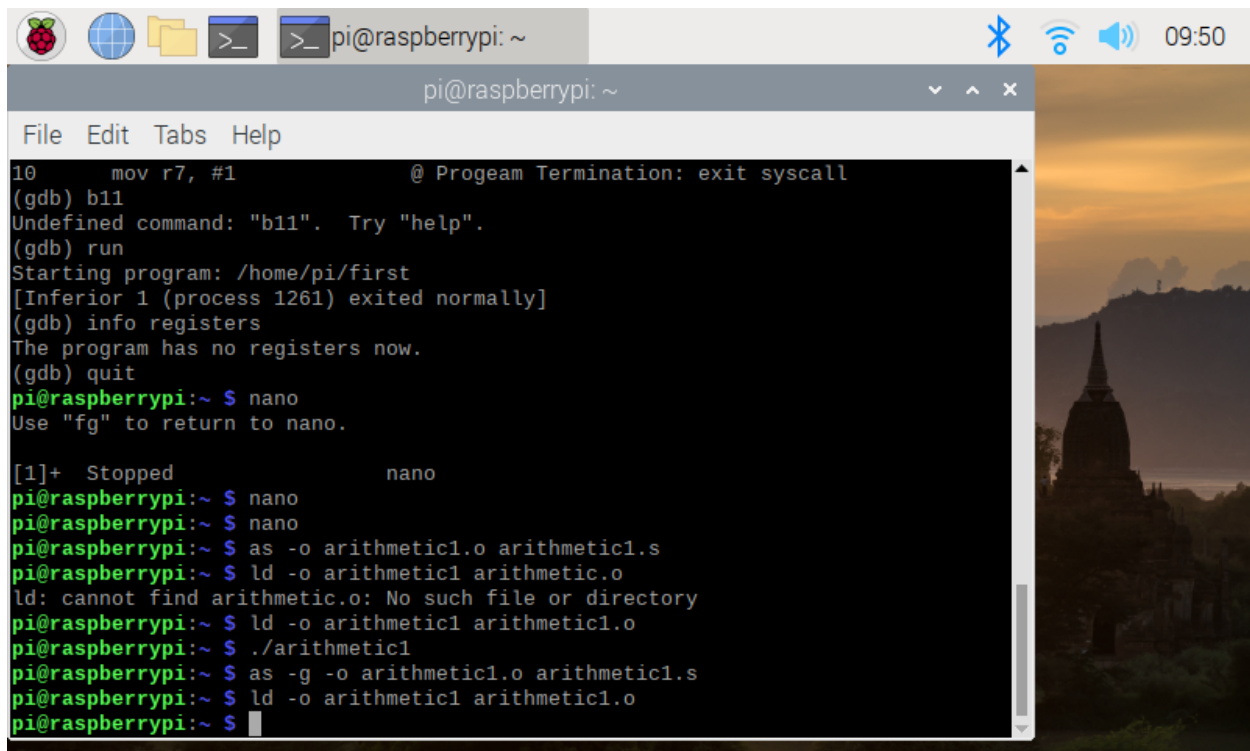
Computer Org & Prog

Project A1

Wrote the program and as instructed.

A screenshot of a terminal window on a Raspberry Pi. The window title is 'pi@raspberrypi: ~'. The terminal shows a GDB session. It starts with a prompt '(gdb) list' which displays assembly code for a program named 'first'. The code includes instructions to load a value into register r1, subtract 1, and add 4. It also shows a comment about Progeam Termination. The user then enters '(gdb) b11', which results in an 'Undefined command: "b11". Try "help".' error. Next, the user enters '(gdb) run', which starts the program at '/home/pi/first'. The output shows '[Inferior 1 (process 1261) exited normally]'. Finally, the user enters '(gdb) info registers', which returns 'The program has no registers now.' The terminal window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The background of the terminal window shows a sunset scene with a pagoda.

After writing it and running it, no output was shown because we've only assigned values to registers and we haven't run any instructions to produce an output.

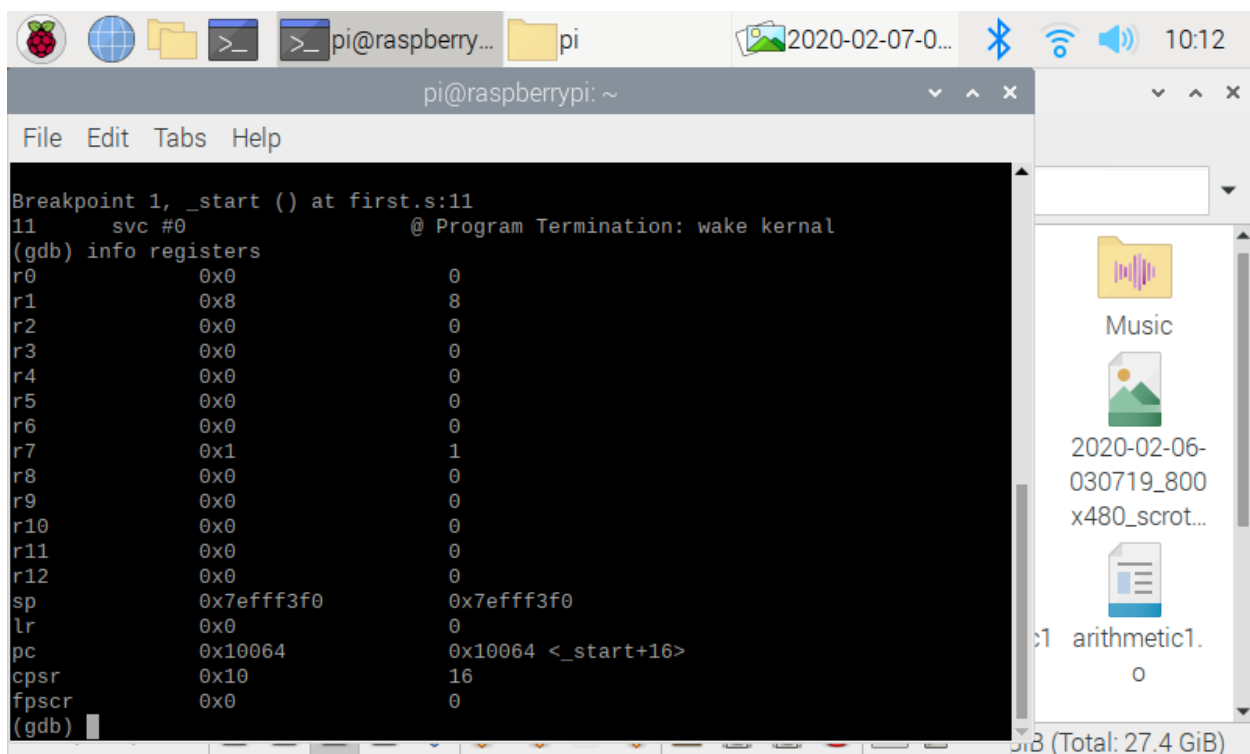


The screenshot shows a terminal window on a Raspberry Pi. The top bar includes system icons and the time 09:50. The terminal window title is 'pi@raspberrypi: ~'. The menu bar shows 'File Edit Tabs Help'. The terminal content shows a GDB session where a program at '/home/pi/first' was run and exited normally. The user then enters 'nano' at the prompt. The nano editor shows a file with assembly code. The user exits nano and returns to the terminal, where they run 'as -o arithmetic1.o arithmetic1.s', 'ld -o arithmetic1 arithmetic1.o', and './arithmetic1'. The terminal shows the linker error 'ld: cannot find arithmetic.o: No such file or directory'.

```
10      mov r7, #1                @ Progeam Termination: exit syscall
(gdb) b11
Undefined command: "b11".  Try "help".
(gdb) run
Starting program: /home/pi/first
[Inferior 1 (process 1261) exited normally]
(gdb) info registers
The program has no registers now.
(gdb) quit
pi@raspberrypi:~ $ nano
Use "fg" to return to nano.

[1]+  Stopped                  nano
pi@raspberrypi:~ $ nano
pi@raspberrypi:~ $ nano
pi@raspberrypi:~ $ as -o arithmetic1.o arithmetic1.s
pi@raspberrypi:~ $ ld -o arithmetic1 arithmetic1.o
ld: cannot find arithmetic.o: No such file or directory
pi@raspberrypi:~ $ ld -o arithmetic1 arithmetic1.o
pi@raspberrypi:~ $ ./arithmetic1
pi@raspberrypi:~ $ as -g -o arithmetic1.o arithmetic1.s
pi@raspberrypi:~ $ ld -o arithmetic1 arithmetic1.o
pi@raspberrypi:~ $
```

Picture is just assembling and taking that to the linker.

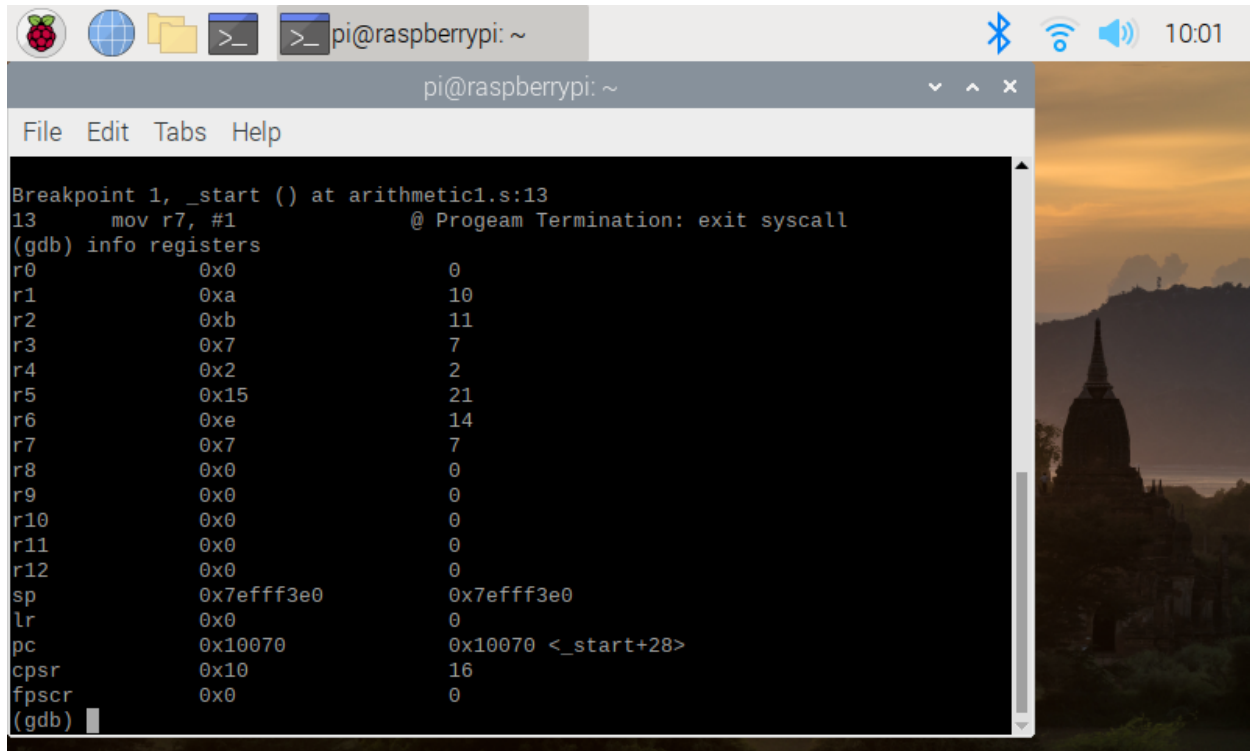


The screenshot shows a terminal window on a Raspberry Pi. The top bar includes system icons and the time 10:12. The terminal window title is 'pi@raspberrypi: ~'. The menu bar shows 'File Edit Tabs Help'. The terminal content shows a GDB session where a breakpoint was set at 'first.s:11'. The user then enters 'info registers'. The terminal shows the register values for r0 through r12, sp, lr, pc, cpsr, and fpscr. The register values are: r0=0x0, r1=0x8, r2=0x0, r3=0x0, r4=0x0, r5=0x0, r6=0x0, r7=0x1, r8=0x0, r9=0x0, r10=0x0, r11=0x0, r12=0x0, sp=0x7efff3f0, lr=0x0, pc=0x10064, cpsr=0x10, fpscr=0x0. The terminal also shows the file 'arithmetic1.o' in the directory.

```
Breakpoint 1, _start () at first.s:11
11      svc #0                    @ Program Termination: wake kernal
(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          0x7efff3f0           0x7efff3f0
lr          0x0                  0
pc          0x10064              0x10064 <_start+16>
cpsr        0x10                16
fpscr       0x0                  0
(gdb)
```

Above picture is one of running "info registers"

Here we see the value placed on r1 as 8 which is what is to be expected since you first assigned the value of 5 into r1 then subtracted 1 away from it which results in 4, but then we added 4 to it which gives the final result in 8.



The screenshot shows a terminal window on a Raspberry Pi. The window title is "pi@raspberrypi: ~". The terminal output shows GDB register information for a program named "arithmetic1.s". The registers and their values are as follows:

Register	Value (Hex)	Value (Dec)
r0	0x0	0
r1	0xa	10
r2	0xb	11
r3	0x7	7
r4	0x2	2
r5	0x15	21
r6	0xe	14
r7	0x7	7
r8	0x0	0
r9	0x0	0
r10	0x0	0
r11	0x0	0
r12	0x0	0
sp	0x7efff3e0	0x7efff3e0
lr	0x0	0
pc	0x10070	0x10070 <_start+28>
cpsr	0x10	16
fpscr	0x0	0

The terminal also shows the following text:

```
Breakpoint 1, _start () at arithmetic1.s:13
13      mov r7, #1          @ Progeam Termination: exit syscall
(gdb) info registers
```

Above picture is one of the program arithmetic1 that we were instructed to write while using first.s as a reference. We see r1 stored with the value of 10, r2 stored with 11, r3 stored with 7 r4 stored with 2. R5 is stored with the value of r1 and r2 added together. R6 is stored with the value of r3 and r4 multiplied together. R7 is stored with the value of r5 subtracted by r6. So our answer is the value that is held in r7. When this program is ran we still see no results because we're only storing values in registers and not executing any output.