Code

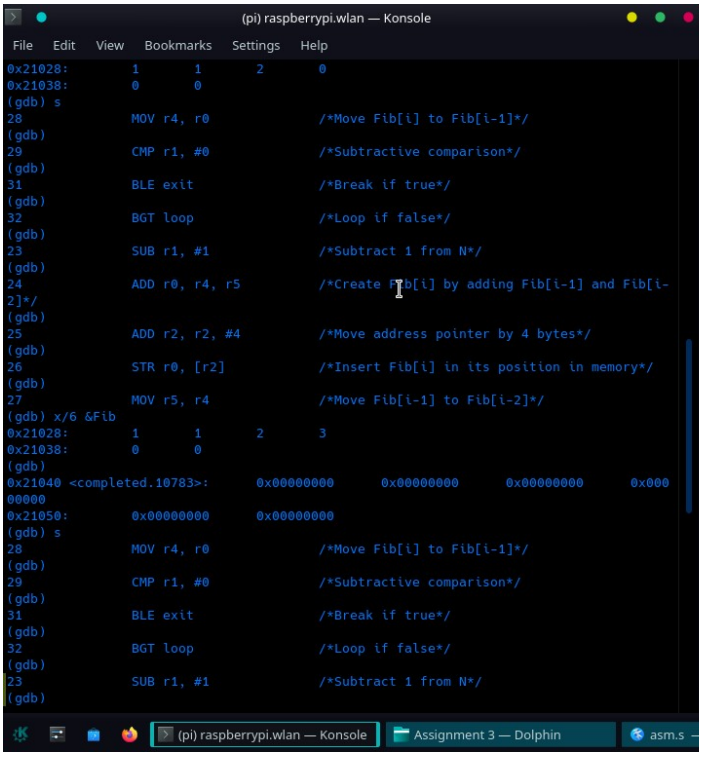
```
.data

.balign 4
N : .word 6
Fib : .word 0,0,0,0,0,0

.text

.global main

main:
        LDR r1, addr_N          /*r1 has the address of N */
        LDR r1, [r1]            /*r1 has the value of N*/
        LDR r2, addr_Fib        /*r2 has the Address of array Fib*/
        MOV r0, #1
        STR r0, [r2]            /*Store 1 on Fib[0]*/
        ADD r2, r2, #4          /*r2 moved to next element Address by 4 bytes*/
        STR r0, [r2]            /*Store 1 on Fib[1]*/
        SUB r1, #2              /*Subtract 2 from N */
        MOV r4, #1              /*Load 1 into r4*/
        MOV r5, #1              /*Load 1 into r5*/
loop:
        SUB r1, #1              /*Subtract 1 from N*/
        ADD r0, r4, r5          /*Create Fib[i] by adding Fib[i-1] and Fib[i-2]*/
        ADD r2, r2, #4          /*Move address pointer by 4 bytes*/
        STR r0, [r2]            /*Insert Fib[i] in its position in memory*/
        MOV r5, r4              /*Move Fib[i-1] to Fib[i-2]*/
        MOV r4, r0              /*Move Fib[i] to Fib[i-1]*/
        CMP r1, #0              /*Subtractive comparison*/
        ITE LE                  /*If/then/else less than or equal to zero*/
        BLE exit                /*Break if true*/
        BGT loop                /*Loop if false*/

exit:
        BX lr

addr_N : .word N
addr_Fib : .word Fib
```

Output

```
Temporary breakpoint 1, main () at asm.s:12
12              LDR r1, addr_N          /*r1 has the address of N */
(gdb) s
13              LDR r1, [r1]            /*r1 has the value of N*/
(gdb)
14              LDR r2, addr_Fib        /*r2 has the Address of array Fib*/
(gdb)
15              MOV r0, #1
(gdb)
16              STR r0, [r2]            /*Store 1 on Fib[0]*/
(gdb) x/6 &Fib
0x21028:        0       0       0       0
0x21038:        0       0
(gdb) s
17              ADD r2, r2, #4          /*r2 moved to next element Address by 4 bytes
*/
(gdb)
18              STR r0, [r2]            /*Store 1 on Fib[1]*/
(gdb)
19              SUB r1, #2              /*Subtract 2 from N */
(gdb) x/6 &Fib
0x21028:        1       1       0       0
0x21038:        0       0
(gdb) s
20              MOV r4, #1              /*Load 1 into r4*/
(gdb)
21              MOV r5, #1              /*Load 1 into r5*/
(gdb)
loop () at asm.s:23
23              SUB r1, #1              /*Subtract 1 from N*/
(gdb)
24              ADD r0, r4, r5          /*Create Fib[i] by adding Fib[i-1] and Fib[i-
2]*/
(gdb)
25              ADD r2, r2, #4          /*Move address pointer by 4 bytes*/
(gdb)
26              STR r0, [r2]            /*Insert Fib[i] in its position in memory*/
(gdb)
```

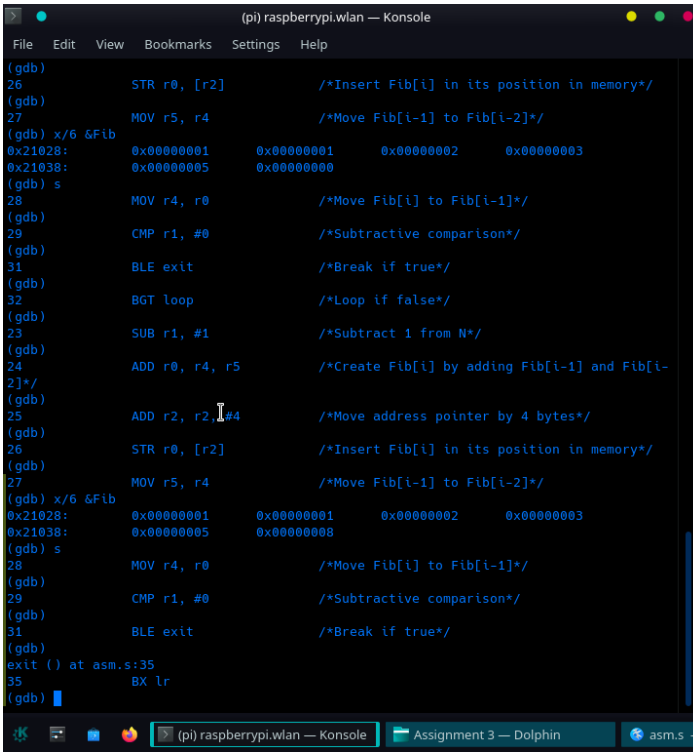
After Fib[0] and Fib[1] are stored

Fib[2] loaded into memory



Fib[3] Loaded into memory

Fib[4] Loaded into memory



Fib[5] Loaded into memory

Program end