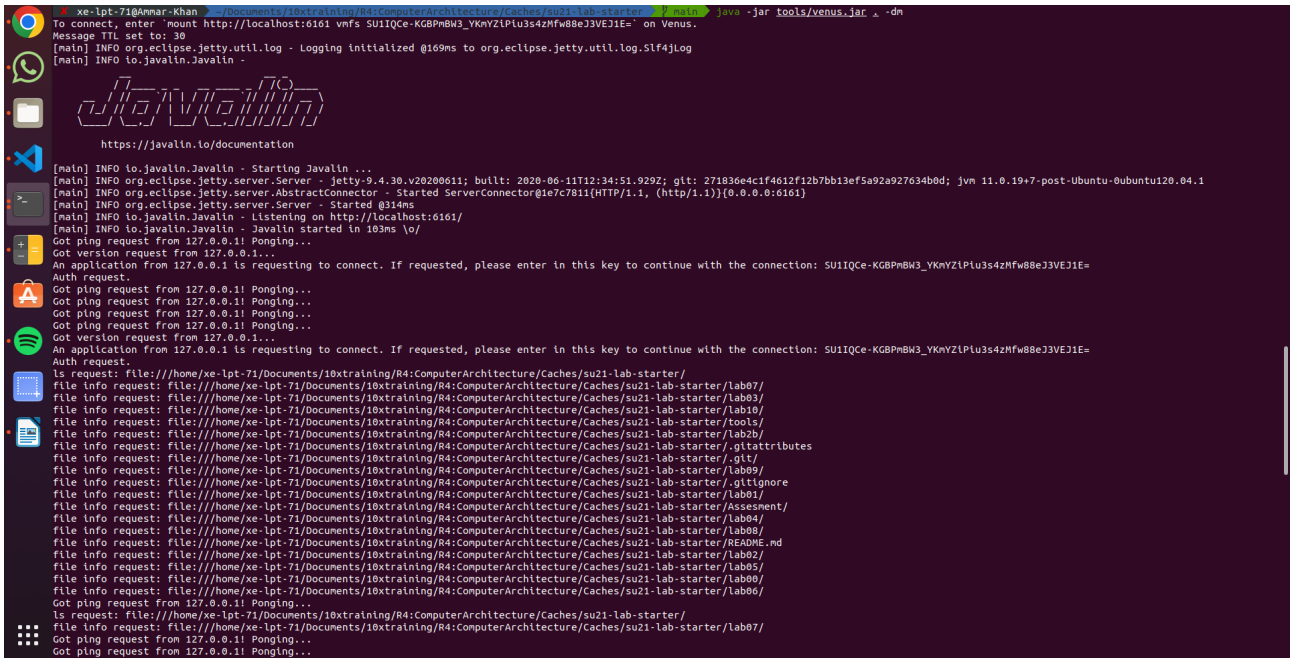
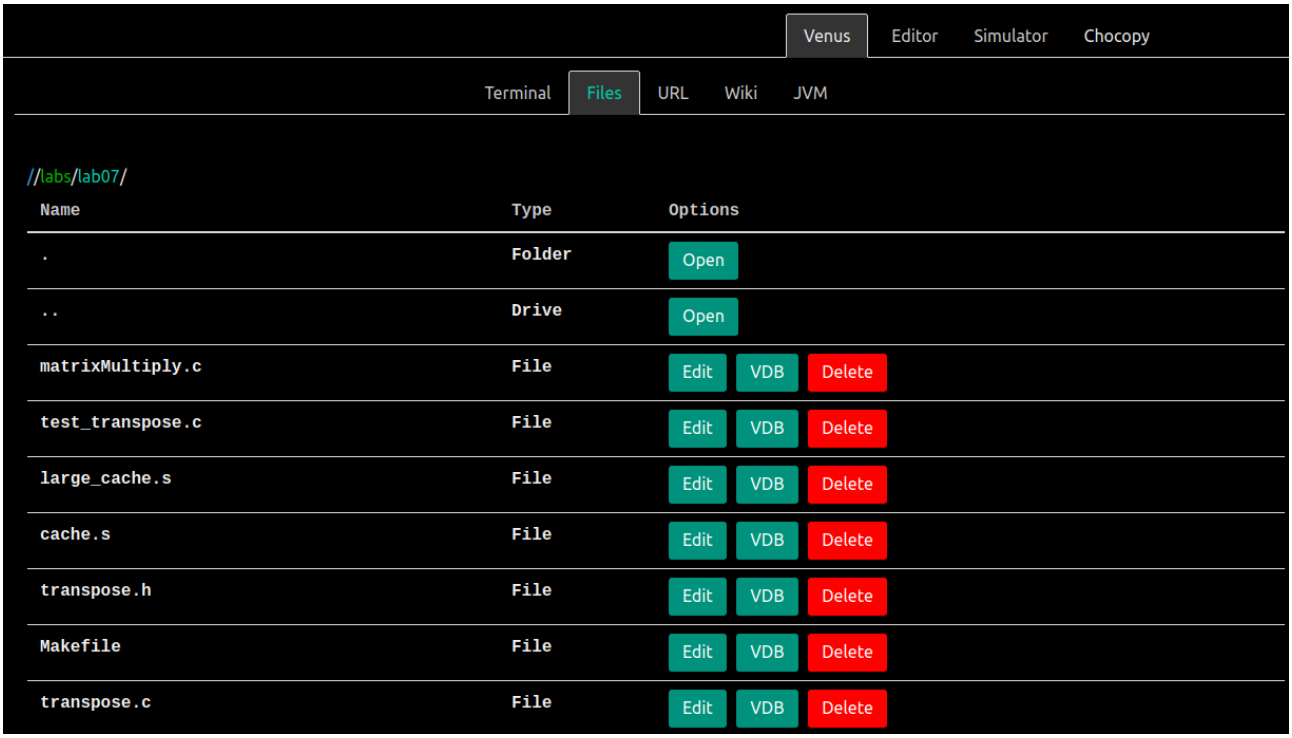
Task 1 : Set up for cache visualizations

1. Venus setup

The venus RISCV simulator was setup successfully the output of venus setup is shown below



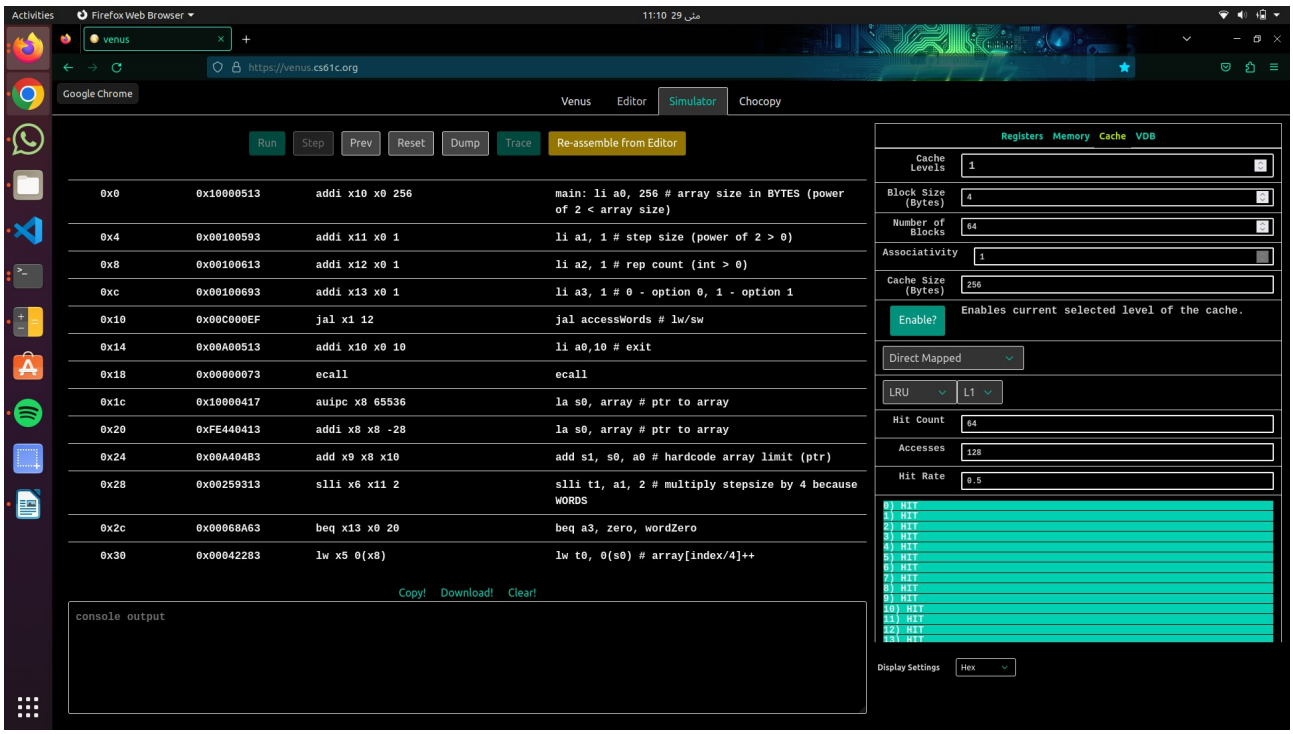


2. The output of running cache.s

I have set the the parameters as shown below

```
18 .data
19 array:  .word   2048                          # max array size specified in BYTES (DO NOT CHANGE)
20
21 .text
22 ###############################################################################################
23 # You MAY change the code below this section
24 main:   li  a0, 256     # array size in BYTES (power of 2 < array size)
25     li  a1, 1       # step size  (power of 2 > 0)
26     li  a2, 1       # rep count  (int > 0)
27     li  a3, 1       # 0 - option 0, 1 - option 1
28 # You MAY change the code above this section
29 ###############################################################################################
```

Since, the array size is 256B, which contains **64 words** in it,the **step size** I have set is **1** which will be multiplied by 4 i.e, the step size is of one **word**. The value of **rep count** set is 1 which means it will be be executed only once unless the condition is false, and **option 1** is selected which will have two cache access read and write so for this code the total number of access would be **128, 64** for read and **64** for write but when the first time read will occur the value would not be present in the cache and there would be **cache miss.** After that the value would be stored in the cache and next time when write operation would occur the value would be already present in the cache and **cache hit** would occur. So there would be of **128** cache access out of which there would be **64** cache hit and **64** cache misses. Hit rate and miss rate would be **50**%. The result got from the venus RISCV simulator is as expected.


Activities
Firefox Web Browser
11:10 29 مئي
venus
https://venus.cs61c.org
Google Chrome
Venus
Editor
Simulator
Chocopy
Run
Step
Prev
Reset
Dump
Trace
Re-assemble from Editor
0x0 0x10000513 addi x10 x0 256 main: li a0, 256 # array size in BYTES (power of 2 < array size)
0x4 0x00100593 addi x11 x0 1 li a1, 1 # step size (power of 2 > 0)
0x8 0x00100613 addi x12 x0 1 li a2, 1 # rep count (int > 0)
0xc 0x00100693 addi x13 x0 1 li a3, 1 # 0 - option 0, 1 - option 1
0x10 0x00C000EF jal x1 12 jal accessWords # lw/sw
0x14 0x00A00513 addi x10 x0 10 li a0,10 # exit
0x18 0x00000073 ecall ecall
0x1c 0x10000417 auipc x8 65536 la s0, array # ptr to array
0x20 0xFE440413 addi x8 x8 -28 la s0, array # ptr to array
0x24 0x00A404B3 add x9 x8 x10 add s1, s0, a0 # hardcode array limit (ptr)
0x28 0x00259313 slli x6 x11 2 slli t1, a1, 2 # multiply stepsize by 4 because WORDS
0x2c 0x00068A63 beq x13 x0 20 beq a3, zero, wordZero
0x30 0x00042283 lw x5 0(x8) lw t0, 0(s0) # array[index/4]++
Copy!
Download!
Clear!
console output
Registers
Memory
Cache
VDB
Cache Levels
1
Block Size (Bytes)
4
Number of Blocks
64
Associativity
1
Cache Size (Bytes)
256
Enable?
Enables current selected level of the cache.
Direct Mapped
LRU
L1
Hit Count
64
Accesses
128
Hit Rate
0.5
0) HIT
1) HIT
2) HIT
3) HIT
4) HIT
5) HIT
6) HIT
7) HIT
8) HIT
9) HIT
10) HIT
11) HIT
12) HIT
Display Settings
Hex