Escuela de Ingeniería en Computadores

CE-1109 Arquitectura de Computadores I

Tarea 2

Estudiante

Mauricio Calderón Chavarría 2019182667

Profesor

Jason Leitón Jiménez

20-08-2024

RISCV

A continuación, se documentan las evidencias de la ejecución del debugging en RISCV así como las capturas de los pasos más importantes durante la instalación de las herramientas.

1. Primeramente, se crea la carpeta central ISA y se procede a crear un directorio solo para el debug de RISCV.

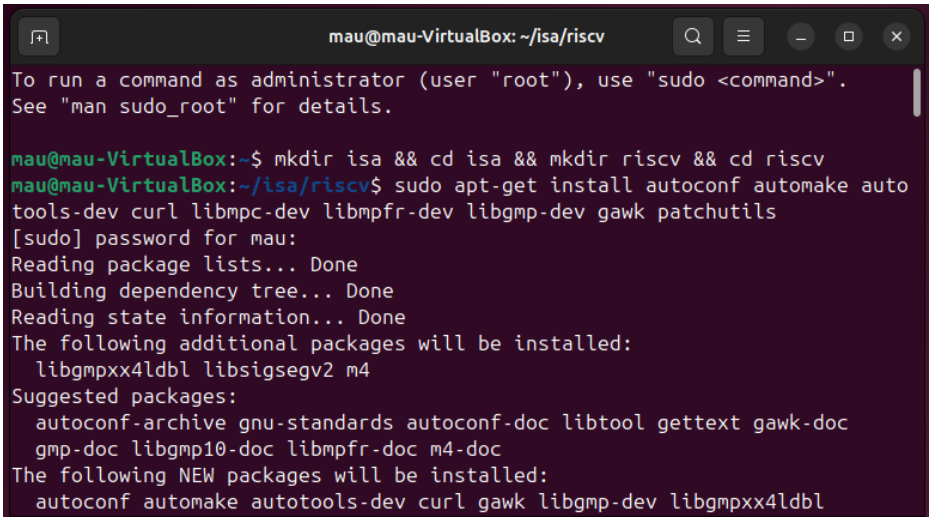


Figura 1. Creación de directorios

2. Seguidamente se ejecutan los comandos para instalar las herramientas necesarias para realizar el debug de RISCV, al ser varios comandos, se observan a continuación en las Figuras 2 y 3.

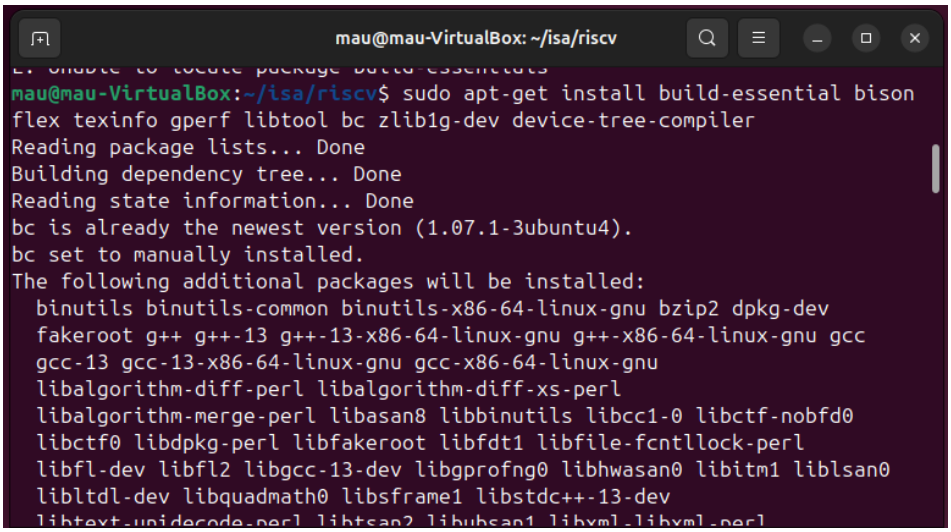


Figura 2. Instalación de los paquetes

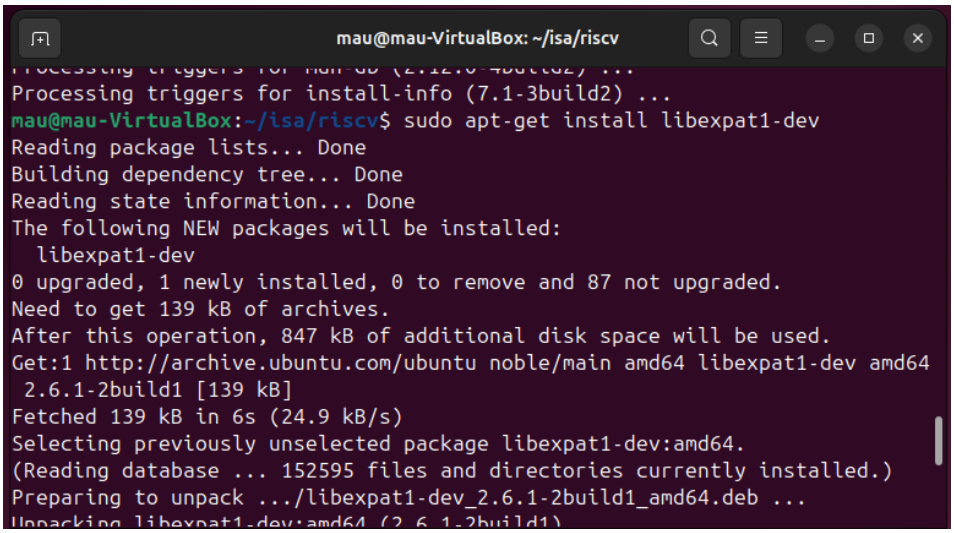


Figura 3. Instalación de los paquetes

3. Se clona el repositorio del toolchain de RISCV.

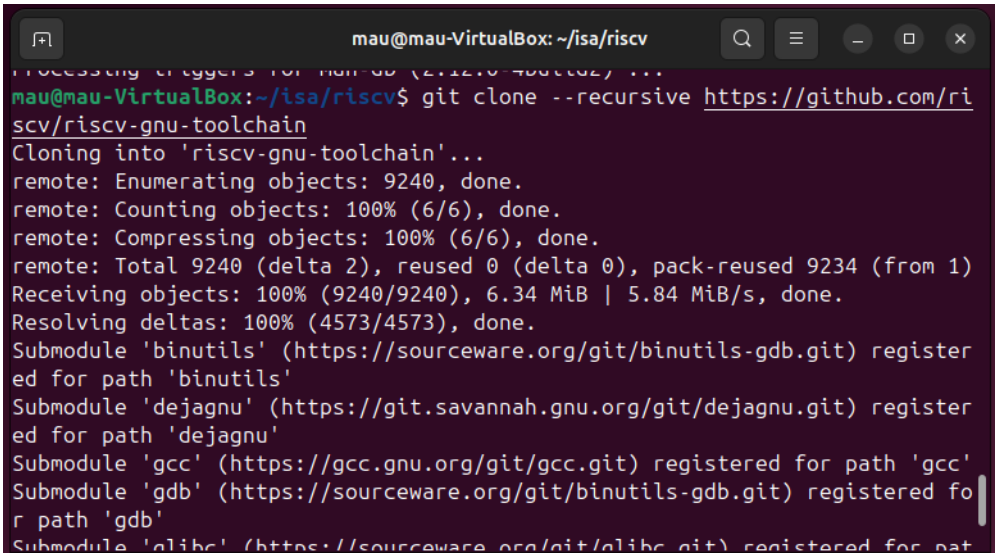


Figura 4. Clonación del toolchain RISCV

4. Ya con la toolchain construida y el RV8 instalado, se procede a realizar el debug del código del triángulo de pascal haciendo uso de las herramientas instaladas. Para este punto ya se creó el archivo .s y el mismo ya posee el código a ejecutar, por lo que se realiza la compilación y enlace, esto se muestra en la Figura 5.

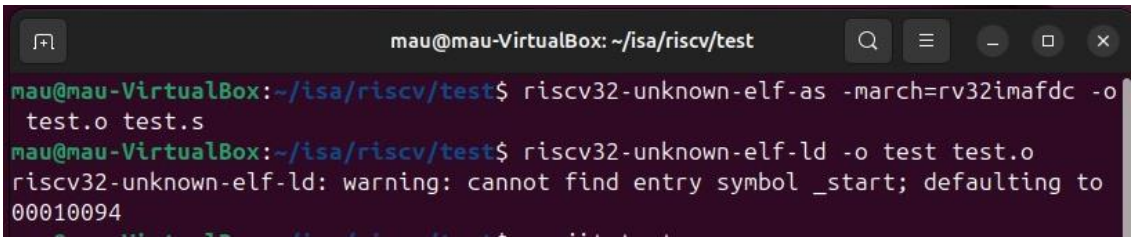


Figura 5. Compilación y enlazamiento.

5. Luego se procede a utilizar el debugger para comprobar el funcionamiento del código. Para esto se puede utilizar el "`rv-jit -d test`" el cual permite establecer `breakpoints`, sin embargo para que se vea un mejor flujo tambien podemos ejecutar simplemente "`rv-jit test`" y con la herramientas nos mostrará el cambio en los registros respectivos. En nuestro caso el que muestra el valor de como se va calculando el triangulo de pascal es t6, el cual es las siguientes imágenes se puede ver como este va conteniendo los valores del triangulo de pascal de un N = 5.

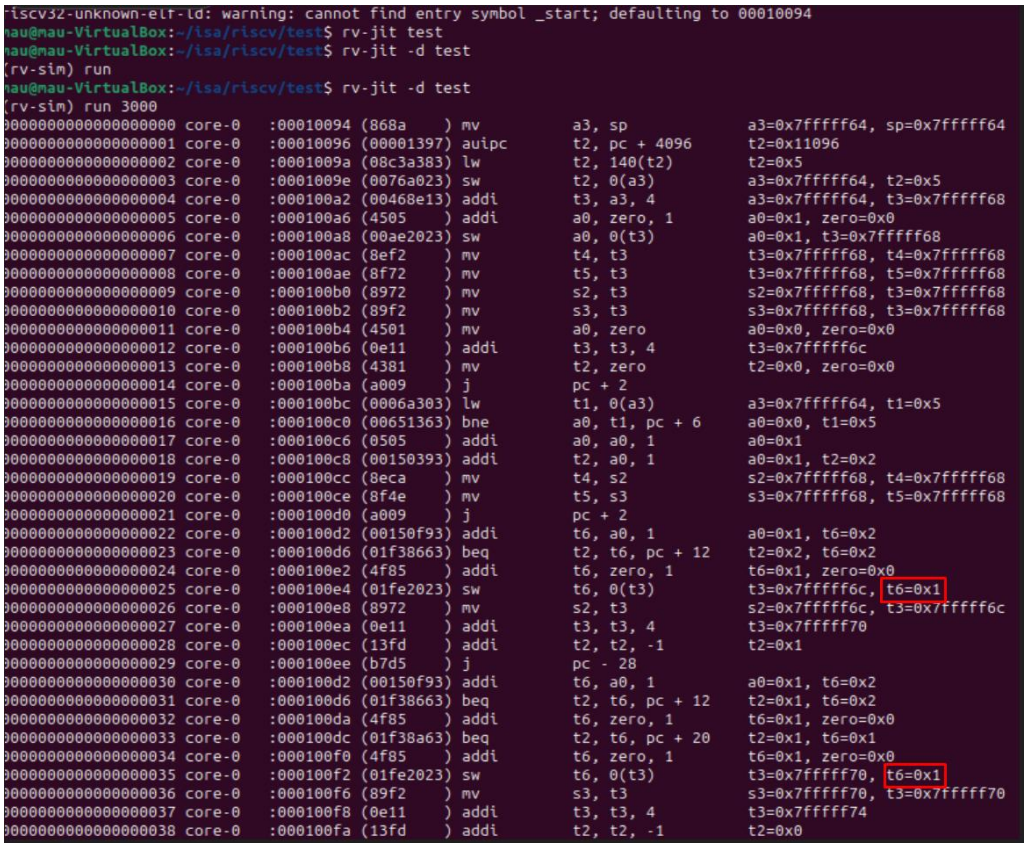


Figura 6. Debug del código en RISCV 1.

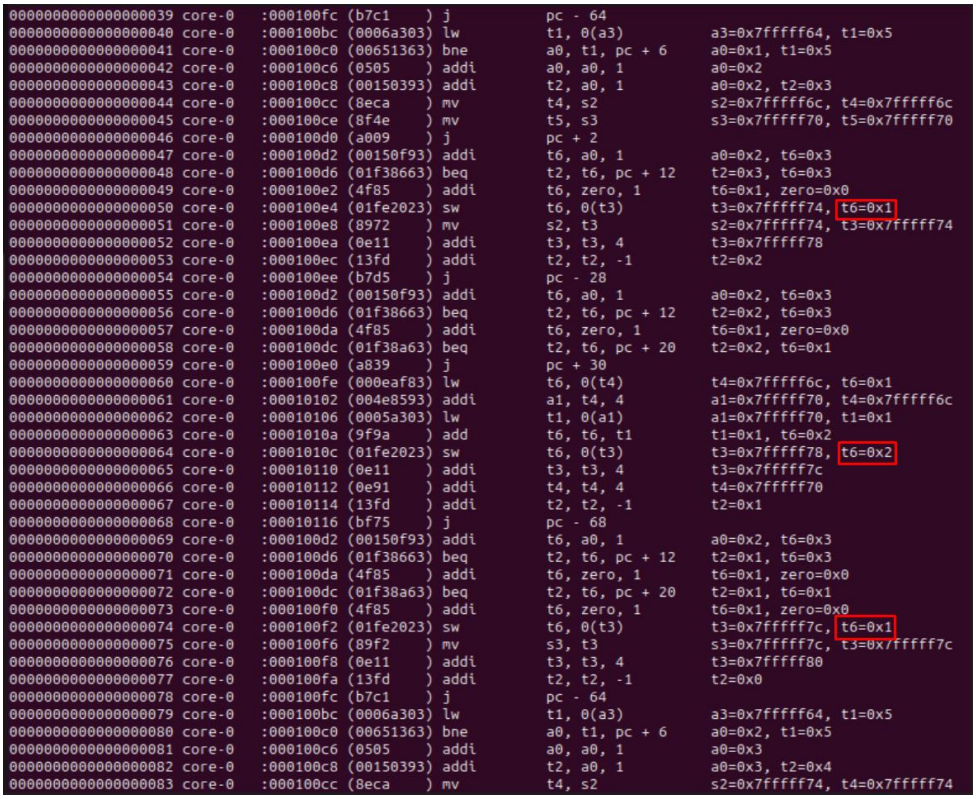


Figura 7. Debug del código en RISCV 2.

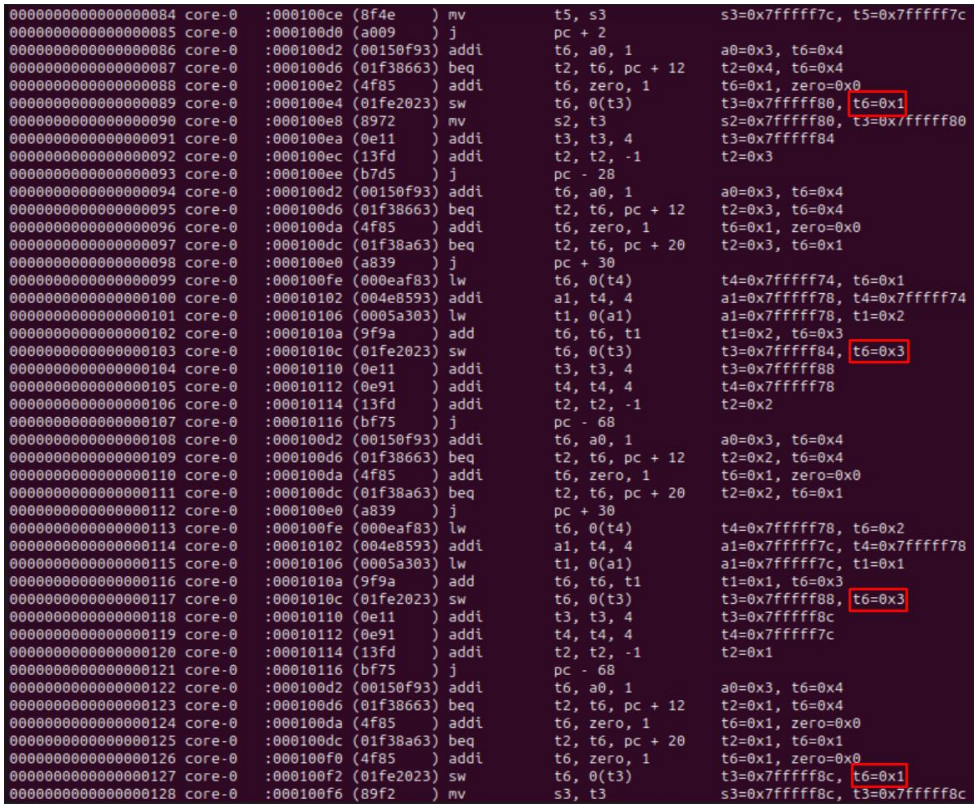
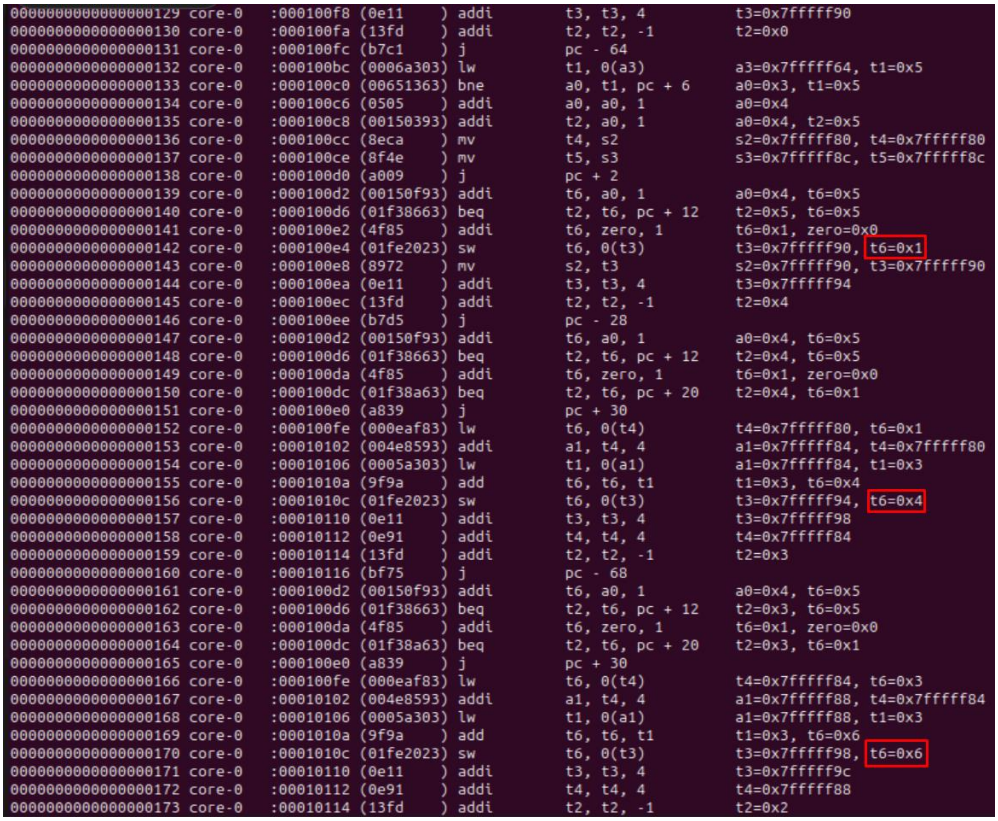
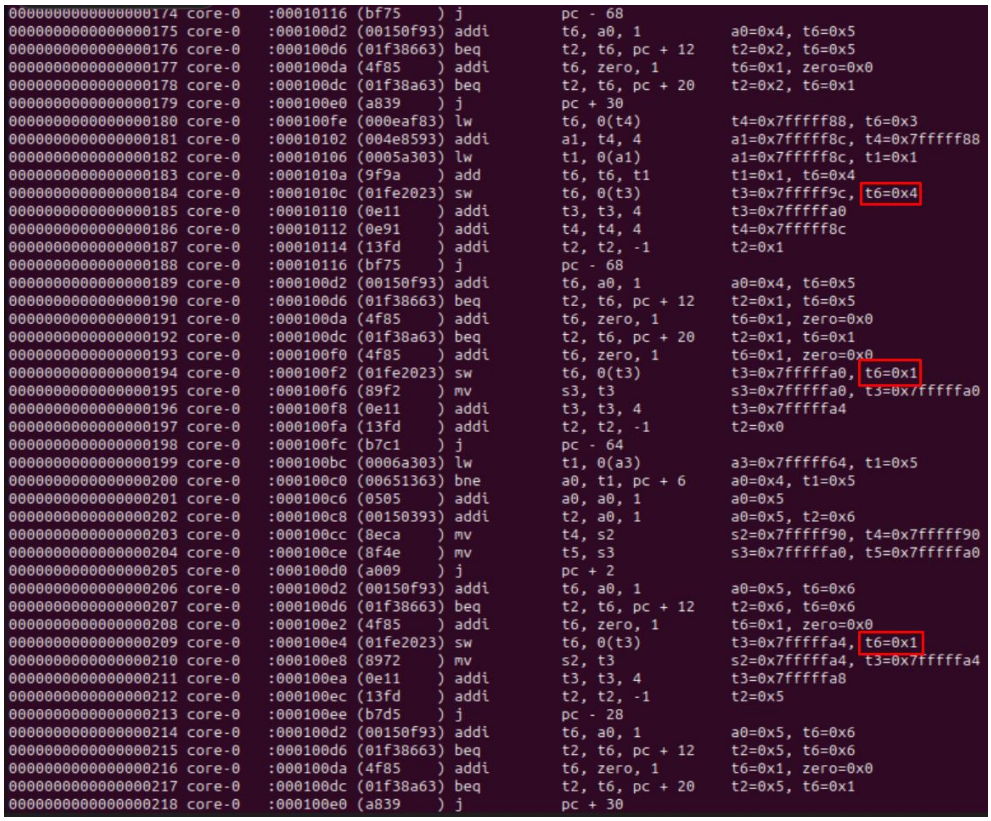


Figura 8. Debug del código en RISCV 3.



```
0000000000000000129 core-0   :000100f8 (0e11    ) addi       t3, t3, 4          t3=0x7fffff90
000000000000000130 core-0   :000100fa (13fd    ) addi       t2, t2, -1         t2=0x0
000000000000000131 core-0   :000100fc (b7c1    ) j          pc - 64
000000000000000132 core-0   :000100bc (0006a303) lw         t1, 0(a3)          a3=0x7fffff64, t1=0x5
000000000000000133 core-0   :000100c0 (00651363) bne        a0, t1, pc + 6     a0=0x3, t1=0x5
000000000000000134 core-0   :000100c6 (0505    ) addi       a0, a0, 1          a0=0x4
000000000000000135 core-0   :000100c8 (00150393) addi       t2, a0, 1          a0=0x4, t2=0x5
000000000000000136 core-0   :000100cc (8eca    ) mv         t4, s2             s2=0x7fffff80, t4=0x7fffff80
000000000000000137 core-0   :000100ce (8f4e    ) mv         t5, s3             s3=0x7fffff8c, t5=0x7fffff8c
000000000000000138 core-0   :000100d0 (a009    ) j          pc + 2
000000000000000139 core-0   :000100d2 (00150f93) addi       t6, a0, 1          a0=0x4, t6=0x5
000000000000000140 core-0   :000100d6 (01f38663) beq        t2, t6, pc + 12    t2=0x5, t6=0x5
000000000000000141 core-0   :000100e2 (4f85    ) addi       t6, zero, 1        t6=0x1, zero=0x0
000000000000000142 core-0   :000100e4 (01fe2023) sw         t6, 0(t3)          t3=0x7fffff90, t6=0x1
000000000000000143 core-0   :000100e8 (8972    ) mv         s2, t3             s2=0x7fffff90, t3=0x7fffff90
000000000000000144 core-0   :000100ea (0e11    ) addi       t3, t3, 4          t3=0x7fffff94
000000000000000145 core-0   :000100ec (13fd    ) addi       t2, t2, -1         t2=0x4
000000000000000146 core-0   :000100ee (b7d5    ) j          pc - 28
000000000000000147 core-0   :000100d2 (00150f93) addi       t6, a0, 1          a0=0x4, t6=0x5
000000000000000148 core-0   :000100d6 (01f38663) beq        t2, t6, pc + 12    t2=0x4, t6=0x5
000000000000000149 core-0   :000100da (4f85    ) addi       t6, zero, 1        t6=0x1, zero=0x0
000000000000000150 core-0   :000100dc (01f38a63) beq        t2, t6, pc + 20    t2=0x4, t6=0x1
000000000000000151 core-0   :000100e0 (a839    ) j          pc + 30
000000000000000152 core-0   :000100fe (000eaf83) lw         t6, 0(t4)          t4=0x7fffff80, t6=0x1
000000000000000153 core-0   :00010102 (004e8593) addi       a1, t4, 4          a1=0x7fffff84, t4=0x7fffff80
000000000000000154 core-0   :00010106 (0005a303) lw         t1, 0(a1)          a1=0x7fffff84, t1=0x3
000000000000000155 core-0   :0001010a (9f9a    ) add        t6, t6, t1         t1=0x3, t6=0x4
000000000000000156 core-0   :0001010c (01fe2023) sw         t6, 0(t3)          t3=0x7fffff94, t6=0x4
000000000000000157 core-0   :00010110 (0e11    ) addi       t3, t3, 4          t3=0x7fffff98
000000000000000158 core-0   :00010112 (0e91    ) addi       t4, t4, 4          t4=0x7fffff84
000000000000000159 core-0   :00010114 (13fd    ) addi       t2, t2, -1         t2=0x3
000000000000000160 core-0   :00010116 (bf75    ) j          pc - 68
000000000000000161 core-0   :000100d2 (00150f93) addi       t6, a0, 1          a0=0x4, t6=0x5
000000000000000162 core-0   :000100d6 (01f38663) beq        t2, t6, pc + 12    t2=0x3, t6=0x5
000000000000000163 core-0   :000100da (4f85    ) addi       t6, zero, 1        t6=0x1, zero=0x0
000000000000000164 core-0   :000100dc (01f38a63) beq        t2, t6, pc + 20    t2=0x3, t6=0x1
000000000000000165 core-0   :000100e0 (a839    ) j          pc + 30
000000000000000166 core-0   :000100fe (000eaf83) lw         t6, 0(t4)          t4=0x7fffff84, t6=0x3
000000000000000167 core-0   :00010102 (004e8593) addi       a1, t4, 4          a1=0x7fffff88, t4=0x7fffff84
000000000000000168 core-0   :00010106 (0005a303) lw         t1, 0(a1)          a1=0x7fffff88, t1=0x3
000000000000000169 core-0   :0001010a (9f9a    ) add        t6, t6, t1         t1=0x3, t6=0x6
000000000000000170 core-0   :0001010c (01fe2023) sw         t6, 0(t3)          t3=0x7fffff98, t6=0x6
000000000000000171 core-0   :00010110 (0e11    ) addi       t3, t3, 4          t3=0x7fffff9c
000000000000000172 core-0   :00010112 (0e91    ) addi       t4, t4, 4          t4=0x7fffff88
000000000000000173 core-0   :00010114 (13fd    ) addi       t2, t2, -1         t2=0x2
```

Figura 9. Debug del código en RISCV 4.



```
000000000000000174 core-0   :00010116 (bf75    ) j          pc - 68
000000000000000175 core-0   :000100d2 (00150f93) addi       t6, a0, 1          a0=0x4, t6=0x5
000000000000000176 core-0   :000100d6 (01f38663) beq        t2, t6, pc + 12    t2=0x2, t6=0x5
000000000000000177 core-0   :000100da (4f85    ) addi       t6, zero, 1        t6=0x1, zero=0x0
000000000000000178 core-0   :000100dc (01f38a63) beq        t2, t6, pc + 20    t2=0x2, t6=0x1
000000000000000179 core-0   :000100e0 (a839    ) j          pc + 30
000000000000000180 core-0   :000100fe (000eaf83) lw         t6, 0(t4)          t4=0x7fffff88, t6=0x3
000000000000000181 core-0   :00010102 (004e8593) addi       a1, t4, 4          a1=0x7fffff8c, t4=0x7fffff88
000000000000000182 core-0   :00010106 (0005a303) lw         t1, 0(a1)          a1=0x7fffff8c, t1=0x1
000000000000000183 core-0   :0001010a (9f9a    ) add        t6, t6, t1         t1=0x1, t6=0x4
000000000000000184 core-0   :0001010c (01fe2023) sw         t6, 0(t3)          t3=0x7fffff9c, t6=0x4
000000000000000185 core-0   :00010110 (0e11    ) addi       t3, t3, 4          t3=0x7ffffa0
000000000000000186 core-0   :00010112 (0e91    ) addi       t4, t4, 4          t4=0x7fffff8c
000000000000000187 core-0   :00010114 (13fd    ) addi       t2, t2, -1         t2=0x1
000000000000000188 core-0   :00010116 (bf75    ) j          pc - 68
000000000000000189 core-0   :000100d2 (00150f93) addi       t6, a0, 1          a0=0x4, t6=0x5
000000000000000190 core-0   :000100d6 (01f38663) beq        t2, t6, pc + 12    t2=0x1, t6=0x5
000000000000000191 core-0   :000100da (4f85    ) addi       t6, zero, 1        t6=0x1, zero=0x0
000000000000000192 core-0   :000100dc (01f38a63) beq        t2, t6, pc + 20    t2=0x1, t6=0x1
000000000000000193 core-0   :000100f0 (4f85    ) addi       t6, zero, 1        t6=0x1, zero=0x0
000000000000000194 core-0   :000100f2 (01fe2023) sw         t6, 0(t3)          t3=0x7ffffa0, t6=0x1
000000000000000195 core-0   :000100f6 (89f2    ) mv         s3, t3             s3=0x7ffffa0, t3=0x7ffffa0
000000000000000196 core-0   :000100f8 (0e11    ) addi       t3, t3, 4          t3=0x7ffffa4
000000000000000197 core-0   :000100fa (13fd    ) addi       t2, t2, -1         t2=0x0
000000000000000198 core-0   :000100fc (b7c1    ) j          pc - 64
000000000000000199 core-0   :000100bc (0006a303) lw         t1, 0(a3)          a3=0x7fffff64, t1=0x5
000000000000000200 core-0   :000100c0 (00651363) bne        a0, t1, pc + 6     a0=0x4, t1=0x5
000000000000000201 core-0   :000100c6 (0505    ) addi       a0, a0, 1          a0=0x5
000000000000000202 core-0   :000100c8 (00150393) addi       t2, a0, 1          a0=0x5, t2=0x6
000000000000000203 core-0   :000100cc (8eca    ) mv         t4, s2             s2=0x7fffff90, t4=0x7fffff90
000000000000000204 core-0   :000100ce (8f4e    ) mv         t5, s3             s3=0x7ffffa0, t5=0x7ffffa0
000000000000000205 core-0   :000100d0 (a009    ) j          pc + 2
000000000000000206 core-0   :000100d2 (00150f93) addi       t6, a0, 1          a0=0x5, t6=0x6
000000000000000207 core-0   :000100d6 (01f38663) beq        t2, t6, pc + 12    t2=0x6, t6=0x6
000000000000000208 core-0   :000100e2 (4f85    ) addi       t6, zero, 1        t6=0x1, zero=0x0
000000000000000209 core-0   :000100e4 (01fe2023) sw         t6, 0(t3)          t3=0x7ffffa4, t6=0x1
000000000000000210 core-0   :000100e8 (8972    ) mv         s2, t3             s2=0x7ffffa4, t3=0x7ffffa4
000000000000000211 core-0   :000100ea (0e11    ) addi       t3, t3, 4          t3=0x7ffffa8
000000000000000212 core-0   :000100ec (13fd    ) addi       t2, t2, -1         t2=0x5
000000000000000213 core-0   :000100ee (b7d5    ) j          pc - 28
000000000000000214 core-0   :000100d2 (00150f93) addi       t6, a0, 1          a0=0x5, t6=0x6
000000000000000215 core-0   :000100d6 (01f38663) beq        t2, t6, pc + 12    t2=0x5, t6=0x6
000000000000000216 core-0   :000100da (4f85    ) addi       t6, zero, 1        t6=0x1, zero=0x0
000000000000000217 core-0   :000100dc (01f38a63) beq        t2, t6, pc + 20    t2=0x5, t6=0x1
000000000000000218 core-0   :000100e0 (a839    ) j          pc + 30
```

Figura 10. Debug del código en RISCV 5.

```
000000000000000219 core-0   :000100fe (000eaf83) lw          t6, 0(t4)          t4=0x7fffff90, t6=0x1
000000000000000220 core-0   :00010102 (004e8593) addi        a1, t4, 4          a1=0x7fffff94, t4=0x7fffff90
000000000000000221 core-0   :00010106 (0005a303) lw          t1, 0(a1)          a1=0x7fffff94, t1=0x4
000000000000000222 core-0   :0001010a (9f9a    ) add         t6, t6, t1         t1=0x4, t6=0x5
000000000000000223 core-0   :0001010c (01fe2023) sw          t6, 0(t3)          t3=0x7ffffa8, t6=0x5
000000000000000224 core-0   :00010110 (0e11    ) addi        t3, t3, 4          t3=0x7fffffac
000000000000000225 core-0   :00010112 (0e91    ) addi        t4, t4, 4          t4=0x7fffff94
000000000000000226 core-0   :00010114 (13fd    ) addi        t2, t2, -1         t2=0x4
000000000000000227 core-0   :00010116 (bf75    ) j           pc - 68
000000000000000228 core-0   :000100d2 (00150f93) addi        t6, a0, 1          a0=0x5, t6=0x6
000000000000000229 core-0   :000100d6 (01f38663) beq         t2, t6, pc + 12    t2=0x4, t6=0x6
000000000000000230 core-0   :000100da (4f85    ) addi        t6, zero, 1        t6=0x1, zero=0x0
000000000000000231 core-0   :000100dc (01f38a63) beq         t2, t6, pc + 20    t2=0x4, t6=0x1
000000000000000232 core-0   :000100e0 (a839    ) j           pc + 30
000000000000000233 core-0   :000100fe (000eaf83) lw          t6, 0(t4)          t4=0x7fffff94, t6=0x4
000000000000000234 core-0   :00010102 (004e8593) addi        a1, t4, 4          a1=0x7fffff98, t4=0x7fffff94
000000000000000235 core-0   :00010106 (0005a303) lw          t1, 0(a1)          a1=0x7fffff98, t1=0x6
000000000000000236 core-0   :0001010a (9f9a    ) add         t6, t6, t1         t1=0x6, t6=0xa
000000000000000237 core-0   :0001010c (01fe2023) sw          t6, 0(t3)          t3=0x7fffffac, t6=0xa
000000000000000238 core-0   :00010110 (0e11    ) addi        t3, t3, 4          t3=0x7ffffb0
000000000000000239 core-0   :00010112 (0e91    ) addi        t4, t4, 4          t4=0x7fffff98
000000000000000240 core-0   :00010114 (13fd    ) addi        t2, t2, -1         t2=0x3
000000000000000241 core-0   :00010116 (bf75    ) j           pc - 68
000000000000000242 core-0   :000100d2 (00150f93) addi        t6, a0, 1          a0=0x5, t6=0x6
000000000000000243 core-0   :000100d6 (01f38663) beq         t2, t6, pc + 12    t2=0x3, t6=0x6
000000000000000244 core-0   :000100da (4f85    ) addi        t6, zero, 1        t6=0x1, zero=0x0
000000000000000245 core-0   :000100dc (01f38a63) beq         t2, t6, pc + 20    t2=0x3, t6=0x1
000000000000000246 core-0   :000100e0 (a839    ) j           pc + 30
000000000000000247 core-0   :000100fe (000eaf83) lw          t6, 0(t4)          t4=0x7fffff98, t6=0x6
000000000000000248 core-0   :00010102 (004e8593) addi        a1, t4, 4          a1=0x7fffff9c, t4=0x7fffff98
000000000000000249 core-0   :00010106 (0005a303) lw          t1, 0(a1)          a1=0x7fffff9c, t1=0x4
000000000000000250 core-0   :0001010a (9f9a    ) add         t6, t6, t1         t1=0x4, t6=0xa
000000000000000251 core-0   :0001010c (01fe2023) sw          t6, 0(t3)          t3=0x7ffffb0, t6=0xa
000000000000000252 core-0   :00010110 (0e11    ) addi        t3, t3, 4          t3=0x7ffffb4
000000000000000253 core-0   :00010112 (0e91    ) addi        t4, t4, 4          t4=0x7fffff9c
000000000000000254 core-0   :00010114 (13fd    ) addi        t2, t2, -1         t2=0x2
000000000000000255 core-0   :00010116 (bf75    ) j           pc - 68
000000000000000256 core-0   :000100d2 (00150f93) addi        t6, a0, 1          a0=0x5, t6=0x6
000000000000000257 core-0   :000100d6 (01f38663) beq         t2, t6, pc + 12    t2=0x2, t6=0x6
000000000000000258 core-0   :000100da (4f85    ) addi        t6, zero, 1        t6=0x1, zero=0x0
000000000000000259 core-0   :000100dc (01f38a63) beq         t2, t6, pc + 20    t2=0x2, t6=0x1
000000000000000260 core-0   :000100e0 (a839    ) j           pc + 30
000000000000000261 core-0   :000100fe (000eaf83) lw          t6, 0(t4)          t4=0x7fffff9c, t6=0x4
000000000000000262 core-0   :00010102 (004e8593) addi        a1, t4, 4          a1=0x7ffffa0, t4=0x7fffff9c
000000000000000263 core-0   :00010106 (0005a303) lw          t1, 0(a1)          a1=0x7ffffa0, t1=0x1
```

Figura 11. Debug del código en RISCV 6.

```
000000000000000259 core-0   :000100dc (01f38a63) beq         t2, t6, pc + 20    t2=0x2, t6=0x1
000000000000000260 core-0   :000100e0 (a839    ) j           pc + 30
000000000000000261 core-0   :000100fe (000eaf83) lw          t6, 0(t4)          t4=0x7fffff9c, t6=0x4
000000000000000262 core-0   :00010102 (004e8593) addi        a1, t4, 4          a1=0x7ffffa0, t4=0x7fffff9c
000000000000000263 core-0   :00010106 (0005a303) lw          t1, 0(a1)          a1=0x7ffffa0, t1=0x1
000000000000000264 core-0   :0001010a (9f9a    ) add         t6, t6, t1         t1=0x1, t6=0x5
000000000000000265 core-0   :0001010c (01fe2023) sw          t6, 0(t3)          t3=0x7ffffb4, t6=0x5
000000000000000266 core-0   :00010110 (0e11    ) addi        t3, t3, 4          t3=0x7ffffb8
000000000000000267 core-0   :00010112 (0e91    ) addi        t4, t4, 4          t4=0x7ffffa0
000000000000000268 core-0   :00010114 (13fd    ) addi        t2, t2, -1         t2=0x1
000000000000000269 core-0   :00010116 (bf75    ) j           pc - 68
000000000000000270 core-0   :000100d2 (00150f93) addi        t6, a0, 1          a0=0x5, t6=0x6
000000000000000271 core-0   :000100d6 (01f38663) beq         t2, t6, pc + 12    t2=0x1, t6=0x6
000000000000000272 core-0   :000100da (4f85    ) addi        t6, zero, 1        t6=0x1, zero=0x0
000000000000000273 core-0   :000100dc (01f38a63) beq         t2, t6, pc + 20    t2=0x1, t6=0x1
000000000000000274 core-0   :000100f0 (4f85    ) addi        t6, zero, 1        t6=0x1, zero=0x0
000000000000000275 core-0   :000100f2 (01fe2023) sw          t6, 0(t3)          t3=0x7ffffb8, t6=0x1
000000000000000276 core-0   :000100f6 (89f2    ) mv          s3, t3             s3=0x7ffffb8, t3=0x7ffffb8
000000000000000277 core-0   :000100f8 (0e11    ) addi        t3, t3, 4          t3=0x7ffffbc
000000000000000278 core-0   :000100fa (13fd    ) addi        t2, t2, -1         t2=0x0
000000000000000279 core-0   :000100fc (b7c1    ) j           pc - 64
000000000000000280 core-0   :000100bc (0006a303) lw          t1, 0(a3)          a3=0x7fffff64, t1=0x5
000000000000000281 core-0   :000100c0 (00651363) bne         a0, t1, pc + 6     a0=0x5, t1=0x5
000000000000000282 core-0   :000100c4 (a891    ) j           pc + 84
000000000000000283 core-0   :00010118 (05d00893) addi        a7, zero, 93       a7=0x5d, zero=0x0
000000000000000284 core-0   :0001011c (4501    ) mv          a0, zero           a0=0x0, zero=0x0
mau@mau-VirtualBox:~/isa/riscv/test$
```

Figura 12. Debug del código en RISCV 7.

X86

En el caso de x86 se instalo GDB para poder realizar el debug respectivo del código, este no presentó ningún problema durante la instalación, luego de esto se crearon las carpetas y el archivo .asm respectivo, el cual también se adjunta con este documento. Seguidamente se realizó la compilación y enlazamiento del código y se procedió a utilizar GDB para comenzar el debug, esto se muestra en la Figura 13.

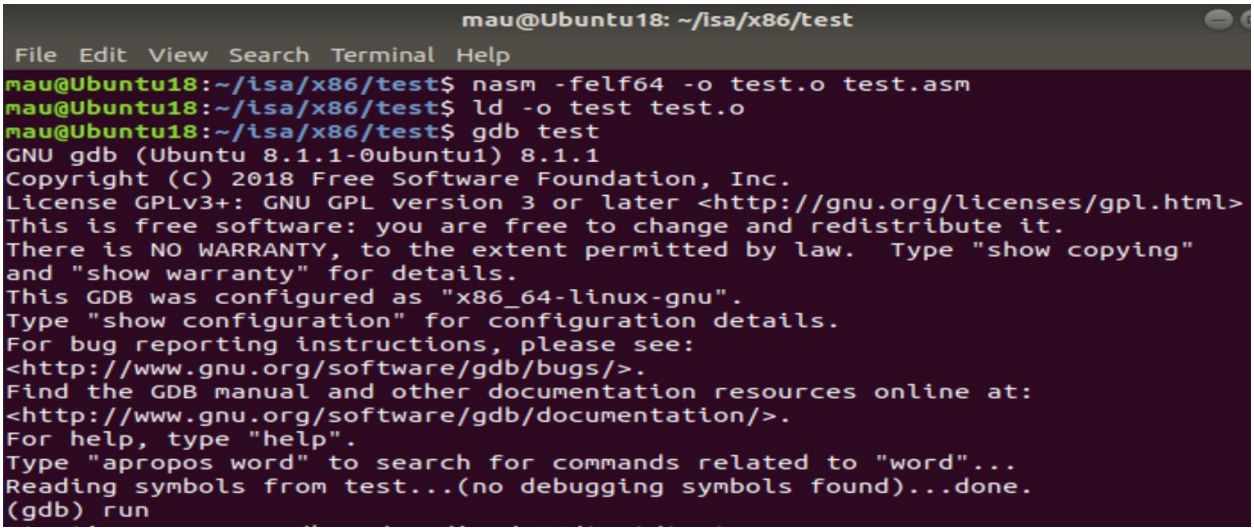


Figura 13. Compilación y enlazamiento del código en x86.

Seguidamente se muestran las capturas del programa en debugging usando GDB

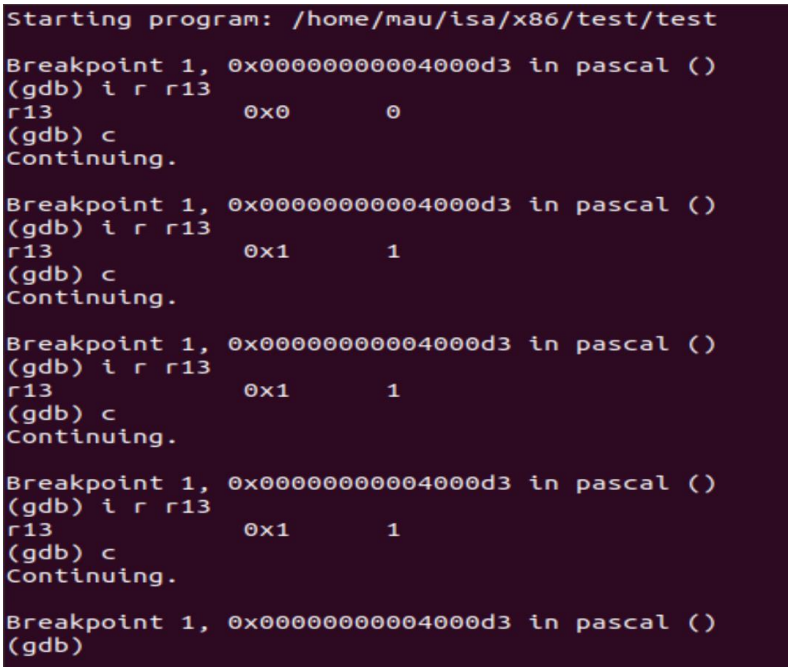


Figura 14. Debug x86 iteración 1

```
Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb) i r r13
r13            0x15     2
(gdb) c
Continuing.

Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb) i r r13
r13            0x1      1
(gdb) c
Continuing.

Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb) i r r13
r13            0x1      1
(gdb) c
Continuing.

Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb) i r r13
r13            0x5      5
(gdb) c
Continuing.

Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb) i r r13
r13            0xd      13
(gdb)
```

Figura 15. Debug x86 iteración 2

```
Continuing.

Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb) i r r13
r13            0x5      5
(gdb) c
Continuing.

Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb) i r r13
r13            0x1      1
(gdb) c
Continuing.

Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb) i r r13
r13            0x1      1
(gdb) c
Continuing.

Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb) i r r13
r13            0x5      5
(gdb) c
Continuing.

Breakpoint 1, 0x00000000004000d3 in pascal ()
(gdb)
```

Figura 16. Debug x86 iteración 3

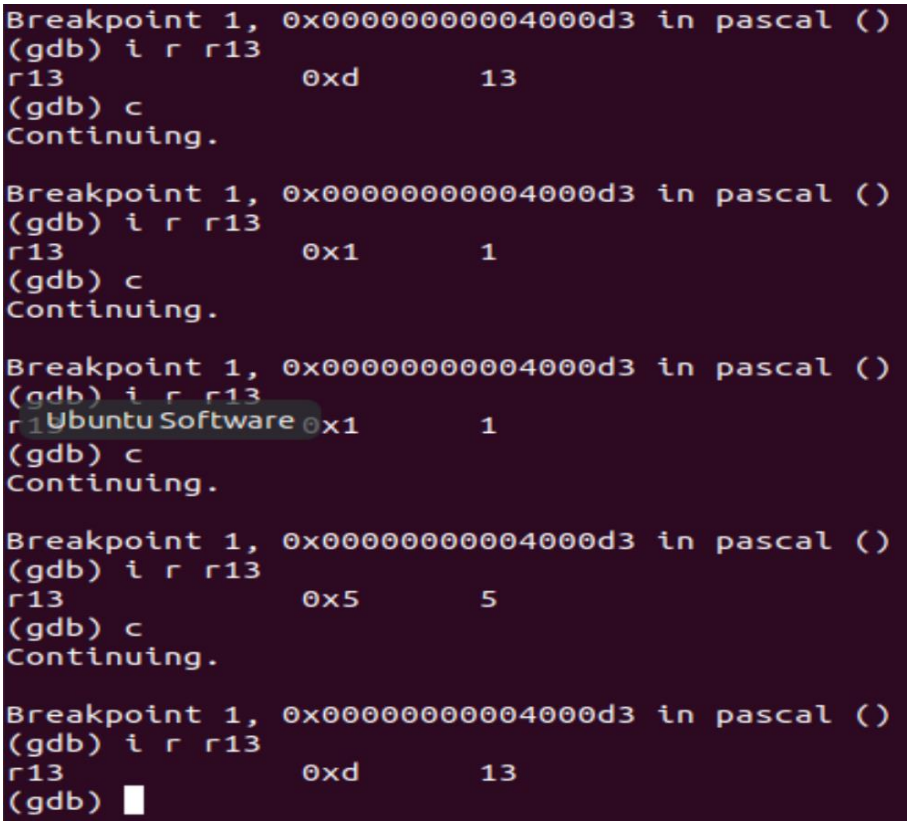


Figura 17. Debug x86 iteración 4

ARM

Por último para ARM se tuvo que realizar utilizando qemu y realizando el debug por sockets, para esto se hizo uso del puerto 1233 y con la ayuda de dos terminales se pudo realizar el debug del código en ARM que fue el original entregado en la tarea 1. A continuacion se muestran las imágenes del debug paso a paso del código en ARM señalando el estado del registro R7 que es el que almacenaba los valores calculados del triángulo de pascal.

Debido a que el breakpoint establecido en la sección pascal, no es posible observar el primer valor de 1 por lo que en la iteración 1 el valor sale como 0, pero en realidad en memoria si se quemo el nivel 0 que es 1.

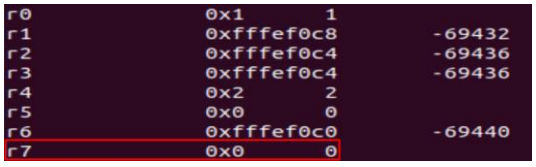


Figura 18 Debug ARM iteración 1

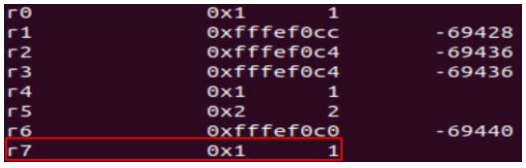


Figura 19 Debug ARM iteración 2

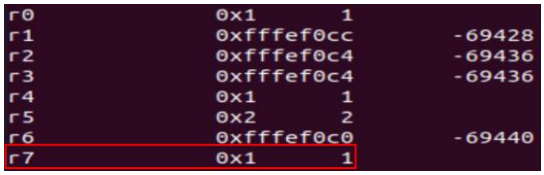


Figura 20 Debug ARM iteración 3

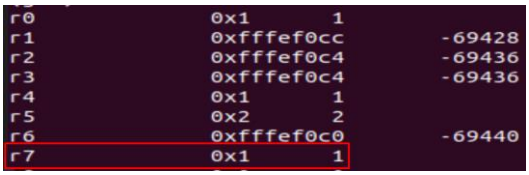


Figura 21 Debug ARM iteración 4

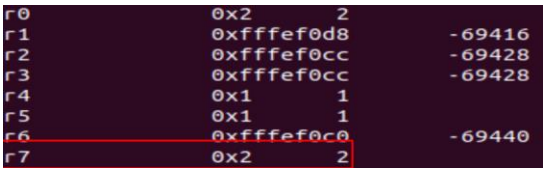


Figura 22 Debug ARM iteración 5

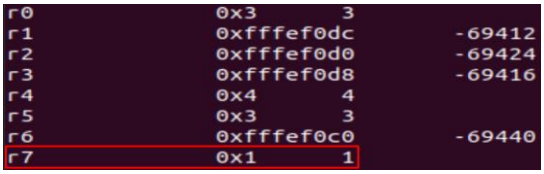


Figura 23 Debug ARM iteración 6

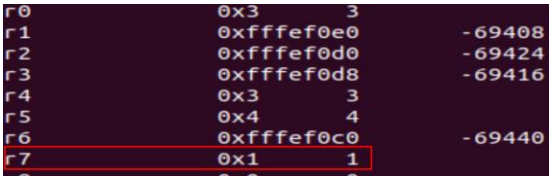


Figura 24 Debug ARM iteración 7

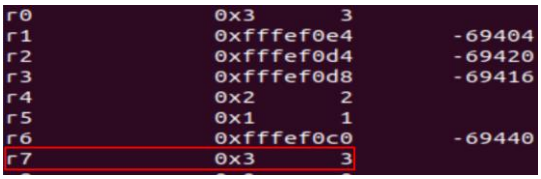


Figura 25 Debug ARM iteración 8

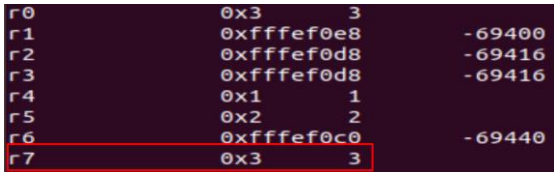


Figura 26 Debug ARM iteración 9

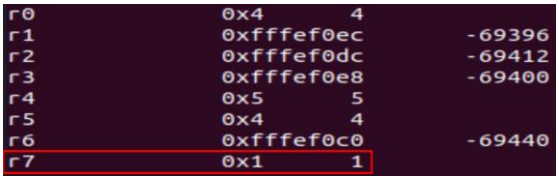


Figura 27 Debug ARM iteración 10

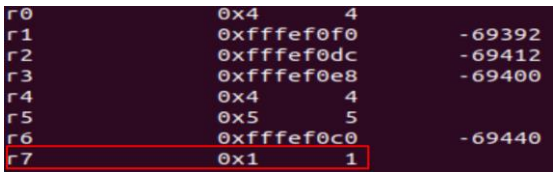


Figura 28 Debug ARM iteración 11

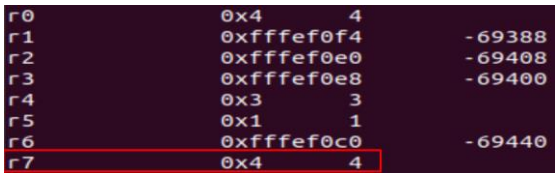


Figura 29 Debug ARM iteración 12

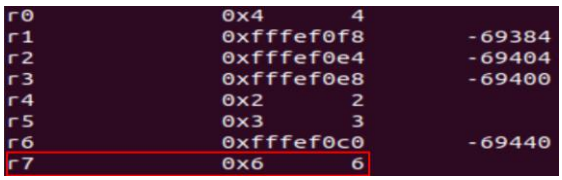


Figura 30 Debug ARM iteración 13

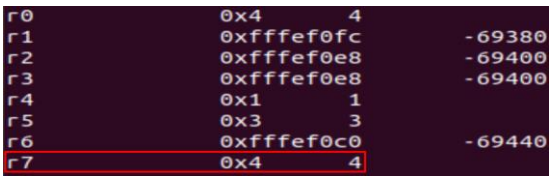


Figura 31 Debug ARM iteración 14

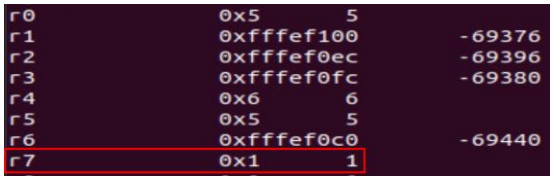


Figura 32 Debug ARM iteración 15

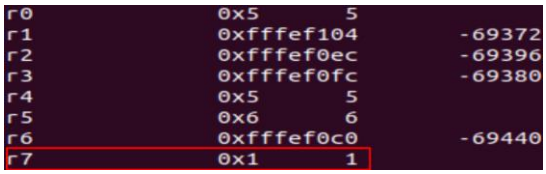


Figura 33 Debug ARM iteración 16

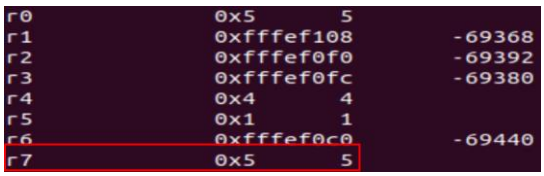


Figura 34 Debug ARM iteración 17

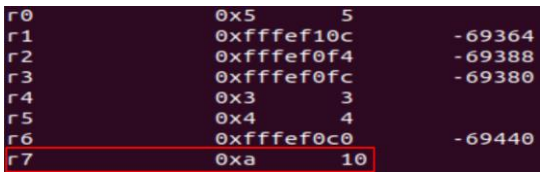


Figura 35 Debug ARM iteración 18

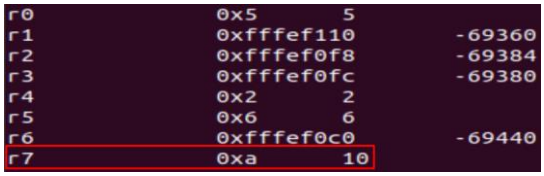


Figura 36 Debug ARM iteración 19

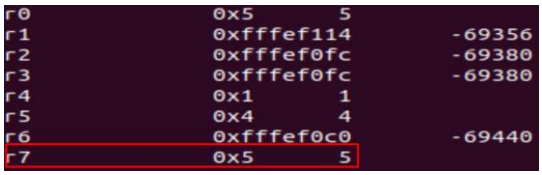


Figura 37 Debug ARM iteración 20

El ultimo 1 no es posible mostrarlo ya que pasa algo parecido al primer 1 el cual no se puede mostrar sin embargo acá tampoco se muestra el registro porque no llega a pasar por la etiqueta “pascal” que es la que muestra los registros sino que sale directamente del programa y muestra este mensaje, confirmando que el código finalizo.

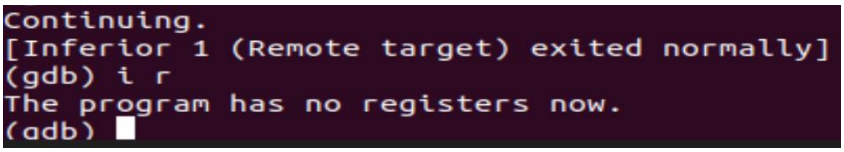


Figura 38 Debug ARM última iteración