

1. Text เป็นการแสดงข้อมูลของ program memory space ซึ่งถูกแบ่งออกเป็น 4 คอร์ลัมตามรูป โดยที่คอร์ลัมแรกเป็นการแสดง memory address ของ instruction นั้น ๆ ด้วยตัวเลขฐาน 16 คอร์ลัมที่สอง นั้นเป็นชุดคำสั่งที่อยู่ใน memory address ซึ่งถูกเขียนด้วยเลขฐาน 16 เป็นชุดคำสั่งที่จะถูกนำไปใช้จริง ๆ บน MIPS ส่วนคอร์ลัมที่สามเป็นชุดคำสั่งที่คนทั่ว ๆ ไปสามารถอ่านออกเขียนได้ และคอร์ลัมสุดท้าย คือภาษา assemble ที่เราเขียนและแทนด้วยสัญลักษณ์

Data	Text
Test	
User Text Segment [00400000]..[00440000]	
[00400000]	8f400000 lw \$4, 0(\$29) ; 193: lw \$a0 0(\$fp) # argc
[00400004]	27a50004 addiu \$5, \$29, 4 ; 194: addiu \$a1 \$fp, 4 # argv
[00400008]	24a60004 addiu \$5, \$5, 4 ; 195: addiu \$a2 \$a1, 4 # nargv
[0040000c]	00041080 sll \$2, \$4, 2 ; 196: sll \$v0 \$a0, 2
[00400010]	00c23021 addu \$6, \$4, \$2 ; 197: addu \$a2 \$a2, \$v0
[00400014]	8c000000 jal 0x00000000 (main) ; 198: jal main
[00400018]	00000000 nop ; 199: nop
[0040001c]	3402000a ori \$2, \$0, 10 ; 19A: li \$v0, 10
[00400020]	0000000c syscall ; 19B: syscall # syscall 10 (exit)
Kernel Text Segment [80000000]..[80010000]	
[80000180]	0001d821 addu \$27, \$0, \$1 ; 90: move \$k1 \$at # Save \$at
[80000184]	3c019000 lui \$1, -28672 ; 92: sw \$v0 \$1 # Not re-entrant and we can't trust \$sp
[80000188]	ac220200 sw \$2, \$12(\$1) ; 93: sw \$a0 \$2 # But we need to use these registers
[8000018c]	3c019000 lui \$1, -28672 ; 95: mfc0 \$k0 \$13 # Cause register
[80000190]	ac240204 sw \$4, \$16(\$1) ; 96: srl \$a0 \$k0, 2 # Extract EnoCode Field
[80000194]	401a6000 mfc0 \$26, \$13 ; 97: andi \$a0 \$a0, 0x1f
[80000198]	001a2082 srl \$4, \$26, 2 ; 101: li \$v0, 4 # syscall 4 (print_str)
[8000019c]	3084001f andi \$4, \$4, 31 ; 102: la \$a0, _ml_
[800001a0]	34020004 ori \$2, \$0, 4 ; 103: syscall
[800001a4]	3c049000 lui \$4, -28672 (_ml_) ; 105: li \$v0, 1 # syscall 1 (print_int)
[800001a8]	0000000c syscall ; 106: srl \$a0 \$k0, 2 # Extract EnoCode Field
[800001ac]	34020001 ori \$2, \$0, 1 ; 107: andi \$a0 \$a0, 0x1f
[800001b0]	001a2082 srl \$4, \$26, 2 ; 108: syscall
[800001b4]	3084001f andi \$4, \$4, 31 ; 110: li \$v0, 4 # syscall 4 (print_str)
[800001b8]	0000000c syscall ; 111: andi \$a0 \$k0, 0x1c
[800001bc]	34020004 ori \$2, \$0, 4 ; 112: lw \$a0, aword(\$a0)
[800001c0]	3344003c andi \$4, \$26, 60 ; 113: lw \$a0, aword(\$a0)
[800001c4]	3c019000 lui \$1, -28672 ; 114: lw \$a0, aword(\$a0)

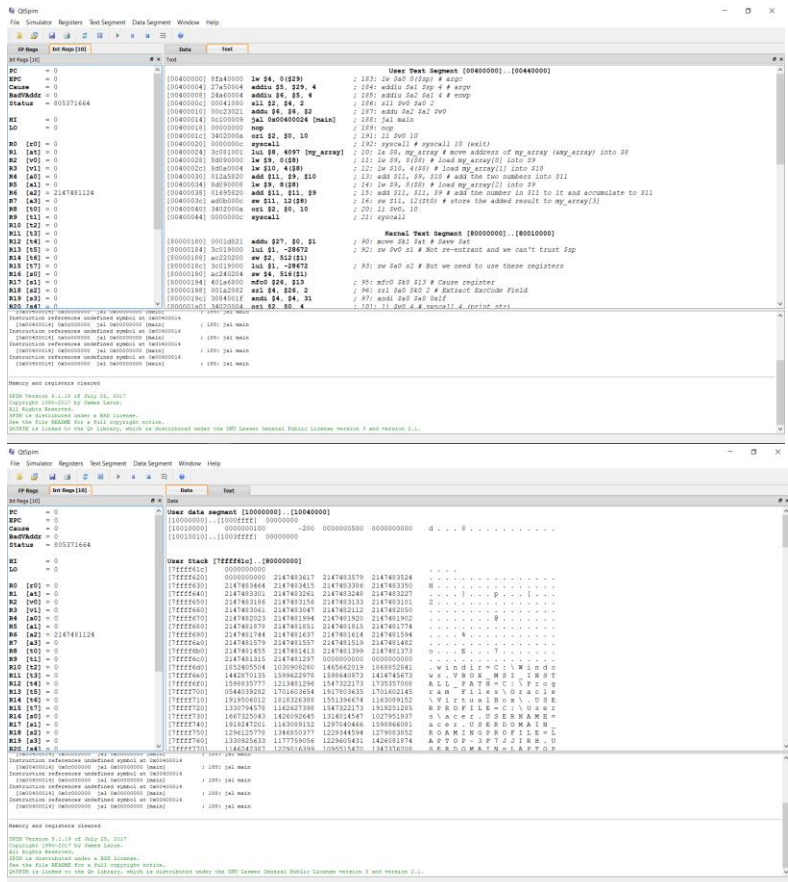
Data นั้นแสดงเกี่ยวกับ Data memory space ซึ่งประกอบไปด้วยตัวแปร ข้อมูล array stack เป็นต้น ซึ่งคอร์ลัมแรกและอันที่สองจะเหมือนกับ tab ของ text ส่วนอันสุดท้ายคือผลลัพธ์

Data	Text
Data	
User data segment [10000000]..[10040000]	
[10000000]..[1003ffff]	00000000
User Stack [7ffff61c]..[80000000]	
[7ffff61c]	0000000000
[7ffff620]	0000000000 2147483617 2147483579 2147483524 - - -
[7ffff630]	2147483464 2147483415 2147483386 2147483350 H - - - - -
[7ffff640]	2147483301 2147483261 2147483240 2147483227 - - - } - - - p - - - i - - -
[7ffff650]	2147483186 2147483156 2147483133 2147483101 2 - - - - -
[7ffff660]	2147483061 2147483047 2147482112 2147482050 - - - - -
[7ffff670]	2147482023 2147481994 2147481920 2147481902 - - - - -
[7ffff680]	2147481878 2147481851 2147481815 2147481774 - - - - -
[7ffff690]	2147481744 2147481637 2147481614 2147481594 - - - - -
[7ffff6a0]	2147481579 2147481557 2147481519 2147481482 - - - - -
[7ffff6b0]	2147481455 2147481413 2147481399 2147481373 0 - - - - -
[7ffff6c0]	2147481315 2147481297 0000000000 0000000000 - - - - -
[7ffff6d0]	1952405504 1030908260 1465662019 1868852841 - - - - -
[7ffff6e0]	1442070135 1599622578 1599640973 1414745673 w s - - - - -
[7ffff6f0]	1598835777 1213481296 1547322173 1735357008 A L L _ P A T H = C : \ p r o g - - - - -
[7ffff700]	0544039282 1701603654 1917803635 1701602145 r a m _ F i l e s \ o r a c l e - - - - -
[7ffff710]	1919506012 1918326386 1551339674 1163091552 V i r t u a l B o x \ _ U S B - - - - -
[7ffff720]	1330794578 1162627398 1547322173 1919251285 R P R O F I L E = C : \ U s e r - - - - -
[7ffff730]	1467325043 1426092645 1314014547 1027951937 s \ a c e r . U S E R N A M E = - - - - -
[7ffff740]	1919247201 1163091552 1297040466 1598964601 a c e r . U S E R D O M A I N - - - - -
[7ffff750]	1296125778 1346850377 1229344594 1279083852 R O A M I N G P R O F I L E = U - - - - -
[7ffff760]	1330925633 1177759056 1229605431 1426081874 A P T O P - 3 F 7 J I R H . U - - - - -
[7ffff770]	1146242387 1229016399 1095515470 1347376208 S E R D O M A I N = L A P T O P - - - - -
[7ffff780]	0927347501 1380534858 1297350728 0977485136 - 3 F 7 J I R H - T M P - c : - - - - -
[7ffff790]	1702057308 1633448818 1551008931 1148219457 \ p e r s o n a l a c e r \ A n d - - - - -

ส่วน int register นั้นเป็นการแสดงค่าสำหรับ integer ของ register สำหรับการใช้งานทั่วไปทั่วไปและมีค่าพิเศษบางอัน

FP Regs [16]	Int Regs [16]	#
[14] Regs [16]		
PC	= 4194324	
EPC	= 0	
Cause	= 0	
BadVAddr	= 0	
Status	= 805371664	
MT	= 0	
LO	= 0	
\$0	(\$0) = 0	
\$1	(\$1) = 0	
\$2	(\$2) = 0	
\$3	(\$3) = 0	
\$4	(\$4) = 0	
\$5	(\$5) = 2147481120	
\$6	(\$6) = 2147481124	
\$7	(\$7) = 0	
\$8	(\$8) = 0	
\$9	(\$9) = 0	
\$10	(\$10) = 0	
\$11	(\$11) = 0	
\$12	(\$12) = 0	
\$13	(\$13) = 0	
\$14	(\$14) = 0	
\$15	(\$15) = 0	
\$16	(\$16) = 0	
\$17	(\$17) = 0	
\$18	(\$18) = 0	
\$19	(\$19) = 0	
\$20	(\$20) = 0	
\$21	(\$21) = 0	
\$22	(\$22) = 0	
\$23	(\$23) = 0	
\$24	(\$24) = 0	
\$25	(\$25) = 0	
\$26	(\$26) = 0	
\$27	(\$27) = 0	
\$28	(\$28) = 0	
\$29	(\$29) = 0	
\$30	(\$30) = 0	
\$31	(\$31) = 0	

2. มีการเอดข้อมูลเพิ่มเข้าไปทั้ง tab ของ text และ tab ของ data ในส่วนของ text นั้นมีการเพิ่มชุดคำสั่งเข้าไปต่อท้ายที่ syscall หลังจากจบชุดคำสั่งก็ปิดท้ายด้วย syscall ส่วนของ text นั้นมีการเพิ่มตรงส่วนของ user data segment เข้ามา และส่วนของ int reg ไม่มีการเปลี่ยนแปลงจากของเดิม



3. ในส่วนของ text และ data นั้นยังคงเดิมอยู่แต่ในส่วนของ int reg นั้นมีการเปลี่ยนแปลงค่าตรง PC และตรงส่วนของตำแหน่ง R ที่ 2, 8, 9, 10, 11, 31

```
PC = 4194372
EPC = 0
Cause = 0
BadVAddr = 0
Status = 805371664

HI = 0
LO = 0

R0 [v0] = 0
R1 [a1] = 0
R2 [v0] = 10
R3 [v1] = 0
R4 [a0] = 1
R5 [a1] = 2147481076
R6 [a2] = 2147481084
R7 [a3] = 0
R8 [v0] = 268500992
R9 [t1] = 500
R10 [t2] = -200
R11 [t3] = 400
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [a0] = 0
R17 [a1] = 0
R18 [a2] = 0
R19 [a3] = 0
R20 [a4] = 0
R21 [a5] = 0
R22 [a6] = 0
R23 [a7] = 0
R24 [t8] = 0
R25 [t9] = 0
R26 [x0] = 0
R27 [x1] = 0
R28 [gp] = 268468224
R29 [sp] = 2147481072
R30 [s8] = 0
R31 [ra] = 4194372
```

4. เพราะว่ามันเป็นการแปลงค่าแบบ signed 2's complement ถึงจะทำให้ค่านั้นกลับไปติดลบเป็นเหมือนเดิม