

# Ejercicio A

## El segmento de Datos (Usuario, Kernel y Pila):

- Usuario

Data	Text
Data	
User data segment [10000000]..[10040000]	
[10000000]..[1000ffff]	00000000
[10010000]	4e495250 49504943 4544204f 4d4f4320 P R I N C I P I O D E C O M
[10010010]	41545550 45524f44 50202e53 2e312052 U T A D O R E S . P R l .
[10010020]	6f47000a 207a656d 6c697547 206e656c . . G o m e z G u i l l e n
[10010030]	61666152 202c6c65 36303659 31353638 R a f a e l , Y 6 0 6 8 6 5 l
[10010040]	61202c54 3130756c 36343130 38373532 T , a l u 0 l 0 l 4 6 2 5 7 8
[10010050]	6c6c7540 7564652e 0a73652e 00000000 @ u l l . e d u . e s . . . . .
[10010060]	490a3d71 4180f159 00000015 0000002c q = . I Y . . A . . . . , . . .
[10010070]	3bafcl0c 4089alcb 00000000 00000000 . . . ; . . . @ . . . . .
[10010080]..[1003ffff]	00000000
User Stack [7ffff834]..[80000000]	
[7ffff834]	00000002 7ffff902 7ffff8e6 . . . . .
[7ffff840]	00000000 7fffffe1 7fffffb3 7ffff82 . . . . .
[7ffff850]	7fffff46 7fffff15 7fffff00 7ffffedc F . . . . .
[7ffff860]	7ffffeaa 7ffffe9d 7ffffe80 7ffffe4f . . . . . O . . . . .
[7ffff870]	7ffffe39 7ffffe22 7ffffdfa 7ffffdca 9 . . . " . . . . .
[7ffff880]	7ffffdbc 7ffffc03 7ffffbc5 7ffffba8 . . . . .
[7ffff890]	7ffffb5e 7ffffb4c 7ffffb34 7ffffb19 ^ . . . L . . . 4 . . . . .
[7ffff8a0]	7ffffafb 7ffffad2 7ffffab4 7ffffa49 . . . . . I . . . . .
[7ffff8b0]	7ffffa32 7ffffa1e 7ffffa0f 7ffff9f9 2 . . . . .
[7ffff8c0]	7ffff9cf 7ffff9a6 7ffff993 7ffff971 . . . . . q . . . . .
[7ffff8d0]	7ffff95b 7ffff939 7ffff927 7ffff912 [ . . . 9 . . . ' . . . . .
[7ffff8e0]	00000000 f3470000 2f7a656d 6e776f44 . . . . . G . m e z / D o w n
[7ffff8f0]	64616f6c 72702f73 69746361 2e316163 l o a d s / p r a c t i c a l .
[7ffff900]	3a430073 6573552f 522f7372 65616661 s . C : / U s e r s / R a f a e
[7ffff910]	5f5f006c 6f4c5350 6f446b63 6f506e77 l . _ _ P S L o c k D o w n P o
[7ffff920]	7963696c 7700303d 69646e69 3a433d72 l i c y = 0 . w i n d i r = C :
[7ffff930]	4e49575c 53574f44 45535500 4f525052 \ W I N D O W S . U S E R P R O
[7ffff940]	454c4946 5c3a433d 72657355 61525c73 F I L E = C : \ U s e r s \ R a
[7ffff950]	6c656166 6df34720 55007a65 4e524553 f a e l G . m e z . U S E R N
[7ffff960]	3d454d41 61666152 47206c65 7a656df3 A M E = R a f a e l G . m e z
[7ffff970]	45535500 4d4f4452 5f4e4941 4d414f52 . U S E R D O M A I N _ R O A M
[7ffff980]	50474e49 49464f52 503d454c 41522d43 I N G P R O F I L E = P C - R A
[7ffff990]	55004146 44524553 49414d4f 43503d4e F A . U S E R D O M A I N = P C
[7ffff9a0]	4641522d 4d540041 3a433d50 6573555c - R A F A . T M P = C : \ U s e
[7ffff9b0]	525c7372 45414641 5c317e4c 44707041 r s \ R A F A E L ~ 1 \ A p p d
[7ffff9c0]	5c617461 61636f4c 65545c6c 5400706d a t a \ L o c a l \ T e m p . T
[7ffff9d0]	3d504d45 555c3a43 73726573 4641525c E M P = C : \ U s e r s \ R A F
[7ffff9e0]	7e4c4541 70415c31 74614470 6f4c5c61 A E L ~ 1 \ A p p D a t a \ L o
[7ffff9f0]	5c6c6163 706d6554 73795300 526d6574 c a l \ T e m p . S y s t e m R
[7ffffa00]	3d746f6f 575c3a43 4f444e49 53005357 o o t = C : \ W I N D O W S . S
[7ffffa10]	65747379 6972446d 433d6576 4553003a y s t e m D r i v e = C : . S E
[7ffffa20]	4f495353 4d414e4e 6f433d45 6c6f736e S S I O N N A M E = C o n s o l
[7ffffa30]	55500065 43494c42 5c3a433d 72657355 e . P U B L I C = C : \ U s e r
[7ffffa40]	75505c73 63696c62 4d535000 6c75646f s \ P u b l i c . P S M o d u l
[7ffffa50]	74615065 3a433d68 6f72505c 6d617267 e P a t h = C : \ P r o g r a m
[7ffffa60]	6c694620 575c7365 6f646e69 6f507377 F i l e s \ W i n d o w s P o
[7ffffa70]	53726577 6c6c6568 646f4d5c 73656c75 w e r S h e l l \ M o d u l e s

- Pila

Data	Text	
Data		
User Stack	[7ffff834]..[80000000]	
[7ffff834]	00000002	7ffff902 7ffff8e6 . . . . .
[7ffff840]	00000000	7fffffe1 7fffffb3 7fffff82 . . . . .
[7ffff850]	7fffff46	7fffff15 7fffff00 7ffffedc F . . . . .
[7ffff860]	7ffffeaa	7ffffe9d 7ffffe80 7ffffe4f . . . . . O . . .
[7ffff870]	7ffffe39	7ffffe22 7ffffdfa 7ffffdca 9 . . . . " . . . . .
[7ffff880]	7ffffdbc	7ffffc03 7ffffbc5 7ffffba8 . . . . .
[7ffff890]	7ffffb5e	7ffffb4c 7ffffb34 7ffffb19 ^ . . . . L . . . . 4 . . . . .
[7ffff8a0]	7ffffafb	7ffffad2 7ffffab4 7ffffa49 . . . . . I . . .
[7ffff8b0]	7ffffa32	7ffffa1e 7ffffa0f 7ffff9f9 2 . . . . .
[7ffff8c0]	7ffff9cf	7ffff9a6 7ffff993 7ffff971 . . . . . q . . .
[7ffff8d0]	7ffff95b	7ffff939 7ffff927 7ffff912 [ . . . . 9 . . . . ' . . . . .
[7ffff8e0]	00000000	f3470000 2f7a656d 6e776f44 . . . . . G . m e z / D o w n
[7ffff8f0]	64616f6c	72702f73 69746361 2e316163 l o a d s / p r a c t i c a l .
[7ffff900]	3a430073	6573552f 522f7372 65616661 s . C : / U s e r s / R a f a e
[7ffff910]	5f5f006c	6f4c5350 6f446b63 6f506e77 l . _ P S L o c k D o w n P o
[7ffff920]	7963696c	7700303d 69646e69 3a433d72 l i c y = 0 . w i n d i r = C :
[7ffff930]	4e49575c	53574f44 45535500 4f525052 \ W I N D O W S . U S E R P R O
[7ffff940]	454c4946	5c3a433d 72657355 61525c73 F I L E = C : \ U s e r s \ R a
[7ffff950]	6c656166	6df34720 55007a65 4e524553 f a e l G . m e z . U S E R N
[7ffff960]	3d454d41	61666152 47206c65 7a656df3 A M E = R a f a e l G . m e z
[7ffff970]	45535500	4d4f4452 5f4e4941 4d414f52 . U S E R D O M A I N _ R O A M
[7ffff980]	50474e49	49464f52 503d454c 41522d43 I N G P R O F I L E = P C - R A
[7ffff990]	55004146	44524553 49414d4f 43503d4e F A . U S E R D O M A I N = P C
[7ffff9a0]	4641522d	4d540041 3a433d50 6573555c - R A F A . T M P = C : \ U s e
[7ffff9b0]	525c7372	45414641 5c317e4c 44707041 r s \ R A F A E L ~ 1 \ A p p d
[7ffff9c0]	5c617461	61636f4c 65545c6c 5400706d a t a \ L o c a l \ T e m p . T
[7ffff9d0]	3d504d45	555c3a43 73726573 4641525c E M P = C : \ U s e r s \ R A F
[7ffff9e0]	7e4c4541	70415c31 74614470 6f4c5c61 A E L ~ 1 \ A p p D a t a \ L o
[7ffff9f0]	5c6c6163	706d6554 73795300 526d6574 c a l \ T e m p . S y s t e m R
[7ffffa00]	3d746f6f	575c3a43 4f444e49 53005357 o o t = C : \ W I N D O W S . S
[7ffffa10]	65747379	6972446d 433d6576 4553003a y s t e m D r i v e = C : . S E
[7ffffa20]	4f495353	4d414e4e 6f433d45 6c6f736e S S I O N N A M E = C o n s o l
[7ffffa30]	55500065	43494c42 5c3a433d 72657355 e . P U B L I C = C : \ U s e r
[7ffffa40]	75505c73	63696c62 4d535000 6c75646f s \ P u b l i c . P S M o d u l
[7ffffa50]	74615065	3a433d68 6f72505c 6d617267 e P a t h = C : \ P r o g r a m
[7ffffa60]	6c694620	575c7365 6f646e69 6f507377 F i l e s \ W i n d o w s P o
[7ffffa70]	53726577	6c6c6568 646f4d5c 73656c75 w e r S h e l l \ M o d u l e s
[7ffffa80]	5c3a433b	444e4957 5c53574f 74737973 ; C : \ W I N D O W S \ s y s t
[7ffffa90]	32336d65	6e69575c 73776f64 65776f50 e m 3 2 \ W i n d o w s P o w e
[7ffffaa0]	65685372	765c6c6c 5c302e31 75646f4d r S h e l l \ v 1 . 0 \ M o d u
[7ffffab0]	0073656c	676f7250 576d6172 32333436 l e s . P r o g r a m W 6 4 3 2
[7ffffac0]	5c3a433d	676f7250 206d6172 656c6946 = C : \ P r o g r a m F i l e
[7ffffad0]	72500073	6172676f 6c69466d 78287365 s . P r o g r a m F i l e s ( x
[7ffffae0]	3d293638	505c3a43 72676f72 46206d61 8 6 ) = C : \ P r o g r a m F
[7ffffaf0]	73656c69	38782820 50002936 72676f72 i l e s ( x 8 6 ) . P r o g r a
[7ffffb00]	69466d61	3d73656c 505c3a43 72676f72 a m F i l e s = C : \ P r o g r a
[7ffffb10]	46206d61	73656c69 6f725000 6d617267 a m F i l e s . P r o g r a m
[7ffffb20]	61746144	5c3a433d 676f7250 446d6172 D a t a = C : \ P r o g r a m D
[7ffffb30]	00617461	434f5250 4f535345 45525f52 a t a . P R O C E S S O R _ R E



- Kernel

Data	Text
Data	
[7ffff30]	46206d61 73656c69 6d6f435c 206e6f6d 206e6f6d am Files \ Common
[7ffff40]	656c6946 6f430073 6e6f6d6d 676f7250 Files . Common Prog
[7ffff50]	466d6172 73656c69 36387828 3a433d29 ram Files ( x 8 6 ) = C :
[7ffff60]	6f72505c 6d617267 6c694620 28207365 \ Program Files (
[7ffff70]	29363878 6d6f435c 206e6f6d 656c6946 x 8 6 ) \ Common File
[7ffff80]	6f430073 6e6f6d6d 676f7250 466d6172 s . Common Program F
[7ffff90]	73656c69 5c3a433d 676f7250 206d6172 iles = C : \ Program
[7ffffa0]	656c6946 6f435c73 6e6f6d6d 6c694620 Files \ Common Fil
[7ffffb0]	41007365 41445050 433d4154 73555c3a es . APPDATA = C : \ Us
[7ffffc0]	5c737265 61666152 47206c65 7a656df3 ers \ Rafael G . me z
[7ffffd0]	7070415c 61746144 616f525c 676e696d \ AppData \ Roaming
[7ffffe0]	4c4c4100 52455355 4f525053 454c4946 . ALLUSERSPROFILE
[7fffff0]	5c3a433d 676f7250 446d6172 00617461 = C : \ Program Data .
Kernel data segment [90000000]..[90010000]	
[90000000]	78452020 74706563 206e6f69 636f2000 Exception . oc
[90000010]	72727563 61206465 6920646e 726f6e67 curred and ignor
[90000020]	000a6465 495b2020 7265746e 74707572 ed . . [ Interrupt
[90000030]	2000205d 4c545b20 20005d42 4c545b20 ] . [ TLB ] . [ TL
[90000040]	20005d42 4c545b20 20005d42 64415b20 B ] . [ TLB ] . [ Ad
[90000050]	73657264 72652073 20726f72 69206e69 dress error in i
[90000060]	2f74736e 61746164 74656620 205d6863 nst / data fetch ]
[90000070]	5b202000 72646441 20737365 6f727265 . [ Address erro
[90000080]	6e692072 6f747320 205d6572 5b202000 r in store ] . [
[90000090]	20646142 74736e69 74637572 206e6f69 Bad instruction
[900000a0]	72646461 5d737365 20200020 6461425b address ] . [ Bad
[900000b0]	74616420 64612061 73657264 00205d73 data address ] .
[900000c0]	455b2020 726f7272 206e6920 63737973 [ Error in sysc
[900000d0]	5d6c6c61 20200020 6572425b 6f706b61 all ] . [ Breakpo
[900000e0]	5d746e69 20200020 7365525b 65767265 int ] . [ Reserve
[900000f0]	6e692064 75727473 6f697463 00205d6e d instruction ] .
[90000100]	5b202000 74697241 74656d68 6f206369 . [ Arithmetic o
[90000110]	66726576 5d776f6c 20200020 6172545b verflow ] . [ Tra
[90000120]	00205d70 5b202000 616f6c46 676e6974 p ] . . [ Floating
[90000130]	696f7020 205d746e 20000000 6f435b20 point ] . . . [ Co
[90000140]	636f7270 005d3220 20000000 444d5b20 proc 2 ] . . . [ MD
[90000150]	005d584d 575b2020 68637461 2020005d MX ] . [ Watch ] .
[90000160]	63614d5b 656e6968 65686320 005d6b63 [ Machine check ] .
[90000170]	00000000 5b202000 68636143 00005d65 . . . . [ Cache ] .
[90000180]	90000024 90000033 9000003b 90000043 \$ . . . 3 . . . ; . . . C . . .
[90000190]	9000004b 90000071 9000008d 900000aa K . . . q . . . . . . . . . .
[900001a0]	900000c0 900000d6 900000e6 90000100 . . . . . . . . . . . . . .
[900001b0]	90000101 9000011a 90000124 90000125 . . . . . . . \$ . . . % . . .
[900001c0]	90000139 9000013a 9000013b 90000148 9 . . . : . . . ; . . . H . . .
[900001d0]	90000149 9000014a 9000014b 90000154 I . . . J . . . K . . . T . . .
[900001e0]	9000015e 90000170 90000171 90000172 ^ . . . p . . . q . . . r . . .
[900001f0]	90000173 90000174 90000175 9000017f s . . . t . . . u . . . . . .
[90000200]..[9000ffff]	00000000

# El segmento de Instrucciones (Usuario y Kernel)

- Usuario

Data	Text
Text	
User Text Segment [00400000]..[00440000]	
[00400000]	8fa40000 lw \$4, 0(\$29) ; 183: lw \$a0 0(\$sp) # argc
[00400004]	27a50004 addiu \$5, \$29, 4 ; 184: addiu \$a1 \$sp 4 # argv
[00400008]	24a60004 addiu \$6, \$5, 4 ; 185: addiu \$a2 \$a1 4 # envp
[0040000c]	00041080 sll \$2, \$4, 2 ; 186: sll \$v0 \$a0 2
[00400010]	00c23021 addu \$6, \$6, \$2 ; 187: addu \$a2 \$a2 \$v0
[00400014]	0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018]	00000000 nop ; 189: nop
[0040001c]	3402000a ori \$2, \$0, 10 ; 191: li \$v0 10
[00400020]	0000000c syscall ; 192: syscall # syscall 10 (exit)
[00400024]	3c011001 lui \$1, 4097 [titulo] ; 6: la \$a0,titulo
[00400028]	34240000 ori \$4, \$1, 0 [titulo]
[0040002c]	34020004 ori \$2, \$0, 4 ; 7: li \$v0,4
[00400030]	0000000c syscall ; 8: syscall
[00400034]	3c011001 lui \$1, 4097 [alumno] ; 11: la \$a0,alumno
[00400038]	34240022 ori \$4, \$1, 34 [alumno]
[0040003c]	34020004 ori \$2, \$0, 4 ; 12: li \$v0,4
[00400040]	0000000c syscall ; 13: syscall
[00400044]	3c011001 lui \$1, 4097 [num1] ; 15: lw \$t0,num1 # carga en el registro \$t0 el valor etiquetado como num1
[00400048]	8c280068 lw \$8, 104(\$1) [num1]
[0040004c]	3c011001 lui \$1, 4097 [num2] ; 16: lw \$t1,num2 # carga en el registro \$t1 el valor etiquetado como num2
[00400050]	8c29006c lw \$9, 108(\$1) [num2]
[00400054]	01095020 add \$10, \$8, \$9 ; 17: add \$t2,\$t0,\$t1 # realiza la siguiente operacion \$t2 = \$t0 + \$t1
[00400058]	014b6020 add \$12, \$10, \$11 ; 19: add \$t4, \$t2, \$t3 # realiza la siguiente operacion \$t4 = \$t2 + \$t3
[0040005c]	340e007d ori \$14, \$0, 125 ; 23: li \$t6,125
[00400060]	11c00004 beq \$14, \$0, 16 [fin_buclewhile-0x00400060]
[00400064]	216bffff addi \$11, \$11, -1 ; 25: addi \$t3,-1
[00400068]	21ceffff addi \$14, \$14, -1 ; 26: addi \$t6,-1
[0040006c]	0401ffff bgez \$0 -12 [buclewhile-0x0040006c]
[00400070]	000b2021 addu \$4, \$0, \$11 ; 32: move \$a0,\$t3
[00400074]	34020001 ori \$2, \$0, 1 ; 33: li \$v0,1
[00400078]	0000000c syscall ; 34: syscall
[0040007c]	3402000a ori \$2, \$0, 10 ; 37: li \$v0,10
[00400080]	0000000c syscall ; 38: syscall
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, 512(\$1)
[8000018c]	3c019000 lui \$1, -28672 ; 93: sw \$a0 \$2 # But we need to use these registers
[80000190]	ac240204 sw \$4, 516(\$1)
[80000194]	401a6800 mfc0 \$26, \$13 ; 95: mfc0 \$k0 \$13 # Cause register
[80000198]	001a2082 srl \$4, \$26, 2 ; 96: srl \$a0 \$k0 2 # Extract ExcCode Field
[8000019c]	3084001f andi \$4, \$4, 31 ; 97: andi \$a0 \$a0 0x1f
[800001a0]	34020004 ori \$2, \$0, 4 ; 101: li \$v0 4 # syscall 4 (print_str)
[800001a4]	3c049000 lui \$4, -28672 [__ml_] ; 102: la \$a0 __ml_
[800001a8]	0000000c syscall ; 103: syscall
[800001ac]	34020001 ori \$2, \$0, 1 ; 105: li \$v0 1 # syscall 1 (print_int)
[800001b0]	001a2082 srl \$4, \$26, 2 ; 106: srl \$a0 \$k0 2 # Extract ExcCode Field

- Kernel

Data	Text
Text	
[00400080]	0000000c syscall ; 38: syscall
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, 512(\$1)
[8000018c]	3c019000 lui \$1, -28672 ; 93: sw \$a0 \$2 # But we need to use these registers
[80000190]	ac240204 sw \$4, 516(\$1)
[80000194]	401a6800 mfc0 \$26, \$13 ; 95: mfc0 \$k0 \$13 # Cause register
[80000198]	001a2082 srl \$4, \$26, 2 ; 96: srl \$a0 \$k0 2 # Extract ExcCode Field
[8000019c]	3084001f andi \$4, \$4, 31 ; 97: andi \$a0 \$a0 0x1f
[800001a0]	34020004 ori \$2, \$0, 4 ; 101: li \$v0 4 # syscall 4 (print_str)
[800001a4]	3c049000 lui \$4, -28672 [__m1_] ; 102: la \$a0 __m1_
[800001a8]	0000000c syscall ; 103: syscall
[800001ac]	34020001 ori \$2, \$0, 1 ; 105: li \$v0 1 # syscall 1 (print_int)
[800001b0]	001a2082 srl \$4, \$26, 2 ; 106: srl \$a0 \$k0 2 # Extract ExcCode Field
[800001b4]	3084001f andi \$4, \$4, 31 ; 107: andi \$a0 \$a0 0x1f
[800001b8]	0000000c syscall ; 108: syscall
[800001bc]	34020004 ori \$2, \$0, 4 ; 110: li \$v0 4 # syscall 4 (print_str)
[800001c0]	3344003c andi \$4, \$26, 60 ; 111: andi \$a0 \$k0 0x3c
[800001c4]	3c019000 lui \$1, -28672 ; 112: lw \$a0 __excp(\$a0)
[800001c8]	00240821 addu \$1, \$1, \$4
[800001cc]	8c240180 lw \$4, 384(\$1)
[800001d0]	00000000 nop ; 113: nop
[800001d4]	0000000c syscall ; 114: syscall
[800001d8]	34010018 ori \$1, \$0, 24 ; 116: bne \$k0 0x18 ok_pc # Bad PC exception requires special checks
[800001dc]	143a0008 bne \$1, \$26, 32 [ok_pc-0x800001dc]
[800001e0]	00000000 nop ; 117: nop
[800001e4]	40047000 mfc0 \$4, \$14 ; 119: mfc0 \$a0 \$14 # EPC
[800001e8]	30840003 andi \$4, \$4, 3 ; 120: andi \$a0 \$a0 0x3 # Is EPC word-aligned?
[800001ec]	10040004 beq \$0, \$4, 16 [ok_pc-0x800001ec]
[800001f0]	00000000 nop ; 122: nop
[800001f4]	3402000a ori \$2, \$0, 10 ; 124: li \$v0 10 # Exit on really bad PC
[800001f8]	0000000c syscall ; 125: syscall
[800001fc]	34020004 ori \$2, \$0, 4 ; 128: li \$v0 4 # syscall 4 (print_str)
[80000200]	3c019000 lui \$1, -28672 [__m2_] ; 129: la \$a0 __m2_
[80000204]	3424000d ori \$4, \$1, 13 [__m2_]
[80000208]	0000000c syscall ; 130: syscall
[8000020c]	001a2082 srl \$4, \$26, 2 ; 132: srl \$a0 \$k0 2 # Extract ExcCode Field
[80000210]	3084001f andi \$4, \$4, 31 ; 133: andi \$a0 \$a0 0x1f
[80000214]	14040002 bne \$0, \$4, 8 [ret-0x80000214]; 134: bne \$a0 0 ret # 0 means exception was an interrupt
[80000218]	00000000 nop ; 135: nop
[8000021c]	401a7000 mfc0 \$26, \$14 ; 145: mfc0 \$k0 \$14 # Bump EPC register
[80000220]	275a0004 addiu \$26, \$26, 4 ; 146: addiu \$k0 \$k0 4 # Skip faulting instruction
[80000224]	409a7000 mtc0 \$26, \$14 ; 148: mtc0 \$k0 \$14
[80000228]	3c019000 lui \$1, -28672 ; 153: lw \$v0 \$1 # Restore other registers
[8000022c]	8c220200 lw \$2, 512(\$1)
[80000230]	3c019000 lui \$1, -28672 ; 154: lw \$a0 \$2
[80000234]	8c240204 lw \$4, 516(\$1)



## El contenido de los Registros Enteros:

FP Reqs	Int Regs [16]	Data	Text
Int Regs [16]		Text	
PC	= 0	[00400080]	0000000c syscall
EPC	= 0		
Cause	= 0		
BadVAddr	= 0		
Status	= 3000fff10		
HI	= 0	[80000180]	0001d821 addu \$27, \$0, \$1
LO	= 0	[80000184]	3c019000 lui \$1, -28672
		[80000188]	ac220200 sw \$2, 512(\$1)
R0 [r0]	= 0	[8000018c]	3c019000 lui \$1, -28672
R1 [at]	= 0	[80000190]	ac240204 sw \$4, 516(\$1)
R2 [v0]	= 0	[80000194]	401a6800 mfc0 \$26, \$13
R3 [v1]	= 0	[80000198]	001a2082 srl \$4, \$26, 2
R4 [a0]	= 2	[8000019c]	3084001f andi \$4, \$4, 31
R5 [a1]	= 7ffff838	[800001a0]	34020004 ori \$2, \$0, 4
R6 [a2]	= 7ffff844	[800001a4]	3c049000 lui \$4, -28672
R7 [a3]	= 0	[800001a8]	0000000c syscall
R8 [t0]	= 0	[800001ac]	34020001 ori \$2, \$0, 1
R9 [t1]	= 0	[800001b0]	001a2082 srl \$4, \$26, 2
R10 [t2]	= 0	[800001b4]	3084001f andi \$4, \$4, 31
R11 [t3]	= 0	[800001b8]	0000000c syscall
R12 [t4]	= 0	[800001bc]	34020004 ori \$2, \$0, 4
R13 [t5]	= 0	[800001c0]	3344003c andi \$4, \$26, 60
R14 [t6]	= 0	[800001c4]	3c019000 lui \$1, -28672
R15 [t7]	= 0	[800001c8]	00240821 addu \$1, \$1, \$4
R16 [s0]	= 0	[800001cc]	8c240180 lw \$4, 384(\$1)
R17 [s1]	= 0	[800001d0]	00000000 nop
R18 [s2]	= 0	[800001d4]	0000000c syscall
R19 [s3]	= 0	[800001d8]	34010018 ori \$1, \$0, 24
R20 [s4]	= 0	[800001dc]	143a0008 bne \$1, \$26, 32
R21 [s5]	= 0	[800001e0]	00000000 nop
R22 [s6]	= 0	[800001e4]	40047000 mfc0 \$4, \$14
R23 [s7]	= 0	[800001e8]	30840003 andi \$4, \$4, 3
R24 [t8]	= 0	[800001ec]	10040004 beq \$0, \$4, 16
R25 [t9]	= 0	[800001f0]	00000000 nop
R26 [k0]	= 0	[800001f4]	3402000a ori \$2, \$0, 10
R27 [k1]	= 0	[800001f8]	0000000c syscall
R28 [gp]	= 10008000	[800001fc]	34020004 ori \$2, \$0, 4
R29 [sp]	= 7ffff834	[80000200]	3c019000 lui \$1, -28672
R30 [s8]	= 0	[80000204]	3424000d ori \$4, \$1, 13
R31 [ra]	= 0	[80000208]	0000000c syscall
		[8000020c]	001a2082 srl \$4, \$26, 2
		[80000210]	3084001f andi \$4, \$4, 31
		[80000214]	14040002 bne \$0, \$4, 8
		[80000218]	00000000 nop
		[8000021c]	401a7000 mfc0 \$26, \$14

## El contenido de los Registros en Punto Flotante:

FP Regs		Int Regs [16]	Data	Text
FP Regs				Text
FIR	= 9800			[00400080] 0000000c syscall
FCSR	= 0			
Single Precision				
FG0	= 0			[80000180] 0001d821 addu \$27, \$0, \$1
FG1	= 0			[80000184] 3c019000 lui \$1, -28672
FG2	= 0			[80000188] ac220200 sw \$2, 512(\$1)
FG3	= 0			[8000018c] 3c019000 lui \$1, -28672
FG4	= 0			[80000190] ac240204 sw \$4, 516(\$1)
FG5	= 0			[80000194] 401a6800 mfc0 \$26, \$13
FG6	= 0			[80000198] 001a2082 srl \$4, \$26, 2
FG7	= 0			[8000019c] 3084001f andi \$4, \$4, 31
FG8	= 0			[800001a0] 34020004 ori \$2, \$0, 4
FG9	= 0			[800001a4] 3c049000 lui \$4, -28672 [
FG10	= 0			[800001a8] 0000000c syscall
FG11	= 0			[800001ac] 34020001 ori \$2, \$0, 1
FG12	= 0			[800001b0] 001a2082 srl \$4, \$26, 2
FG13	= 0			[800001b4] 3084001f andi \$4, \$4, 31
FG14	= 0			[800001b8] 0000000c syscall
FG15	= 0			[800001bc] 34020004 ori \$2, \$0, 4
FG16	= 0			[800001c0] 3344003c andi \$4, \$26, 60
FG17	= 0			[800001c4] 3c019000 lui \$1, -28672
FG18	= 0			[800001c8] 00240821 addu \$1, \$1, \$4
FG19	= 0			[800001cc] 8c240180 lw \$4, 384(\$1)
FG20	= 0			[800001d0] 00000000 nop
FG21	= 0			[800001d4] 0000000c syscall
FG22	= 0			[800001d8] 34010018 ori \$1, \$0, 24
FG23	= 0			[800001dc] 143a0008 bne \$1, \$26, 32 [c
FG24	= 0			[800001e0] 00000000 nop
FG25	= 0			[800001e4] 40047000 mfc0 \$4, \$14
FG26	= 0			[800001e8] 30840003 andi \$4, \$4, 3
FG27	= 0			[800001ec] 10040004 beq \$0, \$4, 16 [ok
FG28	= 0			[800001f0] 00000000 nop
FG29	= 0			[800001f4] 3402000a ori \$2, \$0, 10
FG30	= 0			[800001f8] 0000000c syscall
FG31	= 0			[800001fc] 34020004 ori \$2, \$0, 4
				[80000200] 3c019000 lui \$1, -28672 [
				[80000204] 3424000d ori \$4, \$1, 13 [
				[80000208] 0000000c syscall
				[8000020c] 001a2082 srl \$4, \$26, 2
				[80000210] 3084001f andi \$4, \$4, 31
				[80000214] 14040002 bne \$0, \$4, 8 [ret
				[80000218] 00000000 nop
				[8000021c] 401a7000 mfc0 \$26, \$14
				[80000220] 275a0004 addiu \$26, \$26, 4
				[80000224] 409a7000 mtc0 \$26, \$14
				[80000228] 3c019000 lui \$1, -28672
				[8000022c] 8c220200 lw \$2, 512(\$1)
				[80000230] 3c019000 lui \$1, -28672
				[80000234] 8c240204 lw \$4, 516(\$1)
Double Precision				
FP0	= 0			
FP2	= 0			
FP4	= 0			
FP6	= 0			
FP8	= 0			
FP10	= 0			
FP12	= 0			
FP14	= 0			
FP16	= 0			
FP18	= 0			
FP20	= 0			

## La consola del sistema:



A screenshot of a system console window. The window has a title bar with the text 'Console' and standard window control buttons (minimize, maximize, close). The main area of the window displays the following text:

```
PRINCIPIO DE COMPUTADORES. PR 1.  
Gomez Guillen Rafael, Y6068651T, alu0101462578@ull.edu.es  
-125
```



# Ejercicio B

Sustituye la cadena "apellido1 apellido2 nombre, NIF, alu123456789@ull.edu.es \n" con tu dirección de correo, nombre, apellidos y NIF (NIE o pasaporte). Saca un pantallazo de la consola y marca mediante un cuadro rojo la impresión de tus datos.



```
Console
PRINCIPIO DE COMPUTADORES. PR 1.
Gomez Guillen Rafael, Y6068651T, alu0101462578@ull.edu.es
-125
```

# Ejercicio C

- ¿Qué dirección de memoria (expresa la dirección en hexadecimal) ocupa el primer carácter de tu nombre?

Ocupa la dirección de memoria 10010030.

- ¿Qué carácter es y qué representación tiene en hexadecimal?

Es el carácter R (mayúscula) y en hexadecimal es 52.

User data segment [10000000]..[10040000]					
[10000000]..[1000ffff]	00000000				
[10010000]	4e495250	49504943	4544204f	4d4f4320	P R I N C I P I O D E C O M
[10010010]	41545550	45524f44	50202e53	2e312052	P U T A D O R E S . P R l .
[10010020]	6f47000a	207a656d	6c697547	206e656c	. . G o m e z G u i l l e n
[10010030]	61666152	202c6c65	36303659	31353638	R a f a e l , Y 6 0 6 8 6 5 1
[10010040]	61202c54	3130756c	36343130	38373532	T , a l u 0 1 0 1 4 6 2 5 7 8
[10010050]	6c6c7540	7564652e	0a73652e	00000000	@ u l l . e d u . e s . . . . .
[10010060]	490a3d71	4180f159	00000015	0000002c	q = . I Y . . A . . . . , . . .
[10010070]	3bafc10c	4089a1cb	00000000	00000000	. . . ; . . . @ . . . . .
[10010080]..[1003ffff]	00000000				

- Busca en el segmento de datos de qtspim el número que se encuentra en la dirección etiquetada como num3.

Data	Text				
Data					
User data segment [10000000]..[10040000]					
[10000000]..[1000ffff]	00000000				
[10010000]	4e495250	49504943	4544204f	4d4f4320	P R I N C I P I O D E C O M
[10010010]	41545550	45524f44	50202e53	2e312052	P U T A D O R E S . P R l .
[10010020]	6f47000a	207a656d	6c697547	206e656c	. . G o m e z G u i l l e n
[10010030]	61666152	202c6c65	36303659	31353638	R a f a e l , Y 6 0 6 8 6 5 1
[10010040]	61202c54	3130756c	36343130	38373532	T , a l u 0 1 0 1 4 6 2 5 7 8
[10010050]	6c6c7540	7564652e	0a73652e	00000000	@ u l l . e d u . e s . . . . .
[10010060]	490a3d71	4180f159	00000015	0000002c	q = . I Y . . A . . . . , . . .
[10010070]	3bafc10c	4089a1cb	00000000	00000000	. . . ; . . . @ . . . . .
[10010080]..[1003ffff]	00000000				

- Convierte el número 4.301 a formato IEE-754 para 32 bits.

4.301 a formato IEE-754 para 32 bits es igual a 4089A1CB.

User data segment [10000000]..[10040000]					
[10000000]..[1000ffff]	00000000				
[10010000]	4e495250	49504943	4544204f	4d4f4320	P R I N C I P I O D E C O M
[10010010]	41545550	45524f44	50202e53	2e312052	P U T A D O R E S . P R l .
[10010020]	6f47000a	207a656d	6c697547	206e656c	. . G o m e z G u i l l e n
[10010030]	61666152	202c6c65	36303659	31353638	R a f a e l , Y 6 0 6 8 6 5 1
[10010040]	61202c54	3130756c	36343130	38373532	T , a l u 0 1 0 1 4 6 2 5 7 8
[10010050]	6c6c7540	7564652e	0a73652e	00000000	@ u l l . e d u . e s . . . . .
[10010060]	490a3d71	4180f159	00000015	0000002c	q = . I Y . . A . . . . , . . .
[10010070]	3bafc10c	4089a1cb	00000000	00000000	. . . ; . . . @ . . . . .
[10010080]..[1003ffff]	00000000				

- ¿En qué dirección empieza el número 4.301? expresa la dirección en hexadecimal

Empieza en la dirección 10010074.

- Convierte el número 35531561.13 a formato IEE-754 para 64 bits

4.301 a formato IEE-754 para 64 bits es igual a 4180F159490A3D71.

User data segment [10000000]..[10040000]					
[10000000]..[1000ffff]	00000000				
[10010000]	4e495250	49504943	4544204f	4d4f4320	P R I N C I P I O D E C O M
[10010010]	41545550	45524f44	50202e53	2e312052	P U T A D O R E S . P R 1 .
[10010020]	6f47000a	207a656d	6c697547	206e656c	. . G o m e z G u i l l e n
[10010030]	61666152	202c6c65	36303659	31353638	R a f a e l , Y 6 0 6 8 6 5 1
[10010040]	61202c54	3130756c	36343130	38373532	T , a l u 0 1 0 1 4 6 2 5 7 8
[10010050]	6c6c7540	7564652e	0a73652e	00000000	@ u l l . e d u . e s . . . . .
[10010060]	490a3d71	4180f159	00000015	0000002c	q = . I Y . . A . . . . , . . .
[10010070]	3bafcl0c	4089alcB	00000000	00000000	. . . ; . . . @ . . . . .
[10010080]..[1003ffff]	00000000				

- ¿En qué dirección empieza el número 35531561.13? expresa la dirección en hexadecimal.

Empieza en la dirección 10010064.

## Ejercicio D

- Ejecuta paso a paso el programa hasta que hayas encontrado la instrucción add \$t2,\$t0,\$t1 Una vez se haya ejecutado saca un pantallazo del banco de registros enteros y pon un cuadro rojo sobre el registro \$t2. ¿Qué valor contiene? ¿sabrías expresarlo en decimal?

Tiene un valor igual a 41. En decimal sería 65.

- Cuando hayas terminado de ejecutar esta instrucción, modifica a mano el valor del registro \$t3 (pulsas con el botón derecho del ratón sobre el registro correspondiente en el banco de registro y selecciona "Change Register Contents", allí puedes seleccionar el formato y el valor). Deberás introducir un valor 1200 en formato decimal. Una vez lo hayas hecho saca un pantallazo y marca con un cuadro en rojo el registro correspondiente.



FP Regs		Int Regs [10]	
Int Regs [10]			
PC	=	4194392	
EPC	=	0	
Cause	=	0	
BadVAddr	=	0	
Status	=	805371664	
HI	=	0	
LO	=	0	
R0	[r0]	=	0
R1	[at]	=	268500992
R2	[v0]	=	4
R3	[v1]	=	0
R4	[a0]	=	268501026
R5	[a1]	=	2147481656
R6	[a2]	=	2147481668
R7	[a3]	=	0
R8	[t0]	=	21
R9	[t1]	=	44
R10	[t2]	=	65
R11	[t3]	=	1200
R12	[t4]	=	0
R13	[t5]	=	0
R14	[t6]	=	0
R15	[t7]	=	0
R16	[s0]	=	0
R17	[s1]	=	0
R18	[s2]	=	0
R19	[s3]	=	0
R20	[s4]	=	0
R21	[s5]	=	0
R22	[s6]	=	0
R23	[s7]	=	0
R24	[t8]	=	0
R25	[t9]	=	0
R26	[k0]	=	0
R27	[k1]	=	0
R28	[gp]	=	268468224
R29	[sp]	=	2147481652
R30	[s8]	=	0
R31	[ra]	=	4194328

- A continuación sigue ejecutando paso a paso hasta terminar de ejecutar la instrucción `add $t4,$t2,$t3`. ¿Qué valor tiene el registro `$t4` en hexadecimal? ¿y en decimal?

En hexadecimal es 4F1. Y en decimal es 1265.

- A continuación establece un punto de ruptura “breakpoint” sobre la instrucción `move $a0,$t3` (sobre la instrucción correspondiente, pulsa en el botón derecho del ratón y selecciona “Set Breakpoint”. Después ejecuta todo el código (no paso a paso) y observarás que la ejecución se para en esta instrucción saltándose el bucle que hemos puesto. En este punto. ¿Qué valor tiene \$t3 (expresado en hexadecimal y también en decimal)? ¿y qué valor tiene \$t6?

\$t3 en decimal tiene 1075 y en hexadecimal 433. En cambio, \$t6 en decimal y hexadecimal tiene 0.