

```
1 # Lista.s
2 #
3 # Estructura de datos que implementa una lista doblemente enlazada
4 # con centinela.
5 #
6 # cabeza: nodo centinela.
7 # tamaño: número de elementos de la lista.
8 #
9 # Autores: Ka Fung 18-10492 & Christopher Gomez 18-10892
10 # Fecha: 25-nov-2021
11
12     .data
13
14     .text
15
16 # Funcion crear
17 # Crea una lista circular doblemente enlazada vacia.
18 # Salida:     $v0: lista (negativo si no se pudo crear).
19 #           ($v0): centinela.
20 #           4($v0): tamaño.
21 #
22 # Planificacion de registros:
23 # $t0: dir. de la lista a retornar
24 Lista_crear:
25     # Prologo
26     sw    $fp,    ($sp)
27     move  $fp,    $sp
28     addi  $sp,    $sp, -12
29
30     # Asigna memoria para la lista
31     li    $a0, 8
32     li    $v0, 9
33     syscall
34
35     # Si no me dieron memoria
36     bltz  $v0, Lista_crear_fin
37
38     # Memoria asignada en $t0
39     move  $t0, $v0
40
41     # Reserva memoria para la centinela de la lista
42     li    $a0, 12
43     li    $v0, 9
44     syscall
45
46     # Si hubo error en la creacion del centinela
47     bltz  $v0, Lista_crear_fin
48
49     # La centinela se apunta a si misma
50     sw    $v0,    ($v0)
51     sw    $v0, 8($v0)
52
53     # Inicializa la lista
54     sw    $v0,    ($t0) # nodo cabeza
55     sw    $zero, 4($t0) # tamaño 0
56
57     # Retorna la direccion de la lista
```

```
58     move $v0, $t0
59
60 Lista_crear_fin:
61     # Epilogo
62     move $sp, $fp
63     lw   $fp, ($sp)
64
65     jr $ra
66
67 # Funcion insertar
68 # Inserta un elemento con el valor dado en la lista.
69 # Entrada: $a0: lista.
70 #          $a1: valor del elemento a insertar.
71 # Salida:  $v0: negativo si no se logro insertar
72 #
73 # Planificacion de registros:
74 # $t0: centinela de la lista.
75 # $t1: centinela.anterior.
76 # $t2: tamaño de la lista.
77 # $t3: lista.
78 Lista_insertar:
79     # Prologo
80     sw   $fp, ($sp)
81     move $fp, $sp
82     addi $sp, $sp, -4
83
84     # Guardar la lista en $t3
85     move $t3, $a0
86
87     # Reserva memoria para crear el nodo
88     li $a0, 12
89     li $v0, 9
90     syscall
91
92     # Si hubo error en la creacion del nodo
93     bltz $v0, Lista_insertar_fin
94
95     # Inicializa el valor del nodo
96     sw $a1, 4($v0)
97
98     # Actualiza cabeza y nodo x creado
99     lw $t0, ($t3)
100     lw $t1, ($t0)
101     sw $t1, ($v0) # x.anterior = centinela.anterior
102     sw $t0, 8($v0) # x.siguiete = centinela
103     sw $v0, 8($t1) # centinela.anterior.siguiete = x
104     sw $v0, ($t0) # centinela.anterior = x
105
106     # Actualiza tamaño de la lista
107     lw  $t2, 4($t3)
108     addi $t2, $t2, 1
109     sw  $t2, 4($t3)
110
111 Lista_insertar_fin:
112     # Epilogo
113     move $sp, $fp
114     lw   $fp, ($sp)
```

```
115
116     jr $ra
117
118 # Funcion insertarOrdenado
119 # Inserta un elemento con el valor dado en la lista, manteniendo
120 # un orden indicado por la funcion.
121 #
122 # Entrada: $a0: Lista
123 #          $a1: valor del elemento a insertar.
124 #          $a2: funcion de comparacion.
125 #          (a, b -> bool: 0 si a<b, 1 de otra forma)
126 # Salida: $v0: negativo si no se logro insertar
127 #
128 # Planificacion de registros:
129 # $s0: centinela de la lista.
130 # $s1: nodoAInsertar.
131 # $s2: Lista.
132 # $s3: nodoActual.
133 # $s4: valor del nodo a insertar.
134 # $s5: funcion de comparacion.
135 # $t0: tamaño de la lista.
136 # $t1: nodoActual.anterior.
137 Lista_insertarOrdenado:
138     # Prologo
139     sw $fp, ($sp)
140     sw $ra, -4($sp)
141     sw $s0, -8($sp)
142     sw $s1, -12($sp)
143     sw $s2, -16($sp)
144     sw $s3, -20($sp)
145     sw $s4, -24($sp)
146     sw $s5, -28($sp)
147     move $fp, $sp
148     addi $sp, $sp, -32
149
150     # Si la lista esta vacia se usa el procedimiento insertar
151     lw $t0, 4($a0)
152     bnez $t0, Lista_insertarOrdenado_no_vacia
153
154 Lista_insertarOrdenado_usar_insertar:
155     jal Lista_insertar
156     b Lista_insertarOrdenado_fin
157
158 Lista_insertarOrdenado_no_vacia:
159     # Guardar los argumentos
160     move $s2, $a0
161     move $s4, $a1
162     move $s5, $a2
163
164     # Reserva memoria para crear el nodo
165     li $a0, 12
166     li $v0, 9
167     syscall
168
169     # Si hubo error en la creacion del nodo
170     bltz $v0, Lista_insertarOrdenado_fin
171
```

```
172 # Guarda el nodo e inicializa su valor
173 move $s1, $v0
174 sw $a1, 4($s1)
175
176 # Busca donde insertar el nodo
177 lw $s0, ($s2) # Centinela de la lista
178 lw $s3, 8($s0) # Primer nodo de la lista
179
180 Lista_insertarOrdenado_loop:
181 # while Nodo != centinela
182 beq $s0, $s3, Lista_insertarOrdenado_ultimo
183
184 # Compara nodoActual < nodoAInsertar
185 lw $a0, 4($s3)
186 move $a1, $s4
187 jalr $s5
188
189 # Si nodoActual < nodoAInsertar
190 beqz $v0, Lista_insertarOrdenado_loop_siguiente
191
192 # Si nodoActual >= nodoAInsertar
193 # Inserta el nodo detras de nodoActual
194 lw $t1, ($s3) # nodoActual.anterior
195
196 sw $s1, 8($t1) # nodoActual.anterior.siguiente = nodoAInsertar
197 sw $t1, ($s1) # nodoAInsertar.anterior = nodoActual.anterior
198 sw $s3, 8($s1) # nodoAInsertar.siguiente = nodoActual
199 sw $s1, ($s3) # nodoActual.anterior = nodoAInsertar
200
201 # Actualiza tamaño de la lista
202 lw $t0, 4($s2)
203 addi $t0, $t0, 1
204 sw $t0, 4($s2)
205 b Lista_insertarOrdenado_fin
206
207 Lista_insertarOrdenado_loop_siguiente:
208 # Actualizamos al Nodo.siguiente
209 lw $s3, 8($s3)
210 b Lista_insertarOrdenado_loop
211
212 Lista_insertarOrdenado_ultimo:
213 move $a0, $s2
214 move $a1, $s4
215 b Lista_insertarOrdenado_usar_insertar
216
217 Lista_insertarOrdenado_fin:
218 # Epilogo
219 move $sp, $fp
220 lw $fp, ($sp)
221 lw $ra, -4($sp)
222 lw $s0, -8($sp)
223 lw $s1, -12($sp)
224 lw $s2, -16($sp)
225 lw $s3, -20($sp)
226 lw $s4, -24($sp)
227 lw $s5, -28($sp)
228
```

```
229     jr $ra
230
231 # Funcion eliminar
232 # Elimina el nodo x dado de la lista.
233 # Entrada: $a0: lista.
234 #          $a1: nodo x a eliminar.
235 # Salida: $v0: [0 si se pudo eliminar | 1 de otra manera]
236 #
237 # Planificacion de registros:
238 # $t0: centinela de la lista.
239 # $t1: x.anterior.
240 # $t2: x.siguiente.
241 # $t3: tamaño de la lista.
242 Lista_eliminar:
243     # Prologo
244     sw $fp, ($sp)
245     move $fp, $sp
246     addi $sp, $sp, -4
247
248     # Cargar tamaño de la lista
249     lw $t3, 4($a0)
250
251     li $v0, 0
252     # Si la lista esta vacia
253     beqz $t3, Lista_eliminar_fin
254
255     # Cargar centinela
256     lw $t0, ($a0)
257
258     # Cargar x.anterior y x.siguiente, respectivamente
259     lw $t1, ($a1)
260     lw $t2, 8($a1)
261
262     # Rearreglar apuntadores
263     sw $t2, 8($t1) # x.anterior.siguiente = x.siguiente
264     sw $t1, ($t2) # x.siguiente.anterior = x.anterior
265
266     # Actualizar tamaño
267     addi $t3, $t3, -1
268     sw $t3, 4($a0)
269
270     # Retorna 1
271     li $v0, 1
272
273 Lista_eliminar_fin:
274     # Epilogo
275     move $sp, $fp
276     lw $fp, ($sp)
277
278     jr $ra
279
280 # Funcion eliminarPrimero
281 # Elimina el primer nodo de la lista.
282 # Entrada: $a0: lista.
283 # Salida: $v0: [0 si se pudo eliminar | 1 de otra manera]
284 Lista_eliminarPrimero:
285     # Prologo
```

```
286     sw    $fp,    ($sp)
287     sw    $ra, -4($sp)
288     move  $fp,    $sp
289     addi  $sp,    $sp, -8
290
291     # Cargar centinela.siguiete
292     lw    $a1,    ($a0)
293     lw    $a1, 8($a1)
294
295     jal   Lista_eliminar
296
297     # Epilogo
298     move  $sp,    $fp
299     lw    $fp,    ($sp)
300     lw    $ra, -4($sp)
301
302     jr    $ra
303
304 # Funcion eliminarUltimo
305 # Elimina el ultimo nodo de la lista.
306 # Entrada: $a0: lista.
307 # Salida: $v0: [0 si se pudo eliminar | 1 de otra manera]
308 Lista_eliminarUltimo:
309     # Prologo
310     sw    $fp,    ($sp)
311     sw    $ra, -4($sp)
312     move  $fp,    $sp
313     addi  $sp,    $sp, -8
314
315     # Cargar centinela.anterior
316     lw    $a1, ($a0)
317     lw    $a1, ($a1)
318
319     jal   Lista_eliminar
320
321     # Epilogo
322     move  $sp,    $fp
323     lw    $fp,    ($sp)
324     lw    $ra, -4($sp)
325
326     jr    $ra
327
328 # Funcion primero.
329 # Obtiene el contenido del primer elemento de la lista.
330 # Entrada: $a0: lista.
331 # Salida: $v0: valor del primer elemento de la lista.
332 #
333 # Planificacion de registros:
334 # $t0: tamaño de la lista.
335 Lista_primero:
336     # Prologo
337     sw    $fp,    ($sp)
338     move  $fp,    $sp
339     addi  $sp,    $sp, -4
340
341     li    $v0, -1
342     # Si la lista esta vacia
```

```
343     lw $t0, 4($a0)
344     beqz $t0, Lista_primer_fin
345
346     # Cargar centinela.siguiete
347     lw $a0, ($a0)
348     lw $v0, 8($a0)
349
350     lw $v0, 4($v0)
351
352 Lista_primer_fin:
353     # Epilogo
354     move $sp, $fp
355     lw $fp, ($sp)
356
357     jr $ra
358
359 # Funcion ultimo.
360 # Obtiene el contenido del ultimo elemento de la lista.
361 # Entrada: $a0: lista.
362 # Salida: $v0: valor del primer elemento de la lista.
363 #
364 # Planificacion de registros:
365 # $t0: tamaño de la lista.
366 Lista_ultimo:
367     # Prologo
368     sw $fp, ($sp)
369     move $fp, $sp
370     addi $sp, $sp, -4
371
372     li $v0, -1
373     # Si la lista esta vacia
374     lw $t0, 4($a0)
375     beqz $t0, Lista_ultimo_fin
376
377     # Cargar centinela.anterior
378     lw $a0, ($a0)
379     lw $v0, ($a0)
380
381     lw $v0, 4($v0)
382
383 Lista_ultimo_fin:
384     # Epilogo
385     move $sp, $fp
386     lw $fp, ($sp)
387
388     jr $ra
```