

Sistema operativo				
Nombre clave	Dirección	Instrucción LM	Instrucción LE	Descripción
<i>Nuevo</i>	<i>s0</i>	-1		-1 Es su valor por defecto, el usuario siempre agregará una 8-dir
<i>Actual</i>	<i>s1</i>	6000		Contiene la 6-dir del id del proceso actual, controlado por el lanzador, 6(px)
<i>Validación</i>	<i>s2</i>	1(s1)	LDA s1	El acumulador toma un valor de la forma 6(px)
	<i>s3</i>	3(s98)	BLZ s97	Si acc<0 no hay proceso para actualizar
<i>Actualizar gpc</i>	<i>s4</i>	2000	ADD 000	Se obtiene la 6-dir del gpc del proceso
	<i>s5</i>	6(s7)	STO s7	Se guarda la instrucción 6(px)+1
	<i>s6</i>	1(c1)	LDA C1	Se obtiene el último pc del proceso con forma 8(pc)
	<i>s7</i>	6(p5)	STO p5	En p5 se actualiza gpc
<i>Actualizar gacc</i>	<i>s8</i>	1(s7)	LDA s7	Se obtiene la instrucción 6(px)+1
	<i>s9</i>	2000	ADD 000	Para acceder a la 6-dir del gacc del proceso
	<i>s10</i>	6(s12)	STO s12	En s12 se guarda 6(px)+2
	<i>s11</i>	1(c2)	LDA c2	Se obtiene el último acc del proceso
<i>Actualizar gjump</i>	<i>s12</i>	6(p6)	STO p6	Se actualiza el valor de gacc
	<i>s13</i>	1(s12)	LDA s12	Obtiene la 6(p6) del proceso que se está actualizando
	<i>s14</i>	2000	ADD 000	Obtiene la 6(p7) del proceso que se está ejecutando
	<i>s15</i>	6(s17)	STO s17	Guarda en e18 el código para guardar en la zona de procesos c14
<i>Saltar</i>	<i>s16</i>	1(c14)	LDA c14	Obtiene c14
	<i>s17</i>	6(p7)	STO p7	Guarda en la zona de procesos correspondiente al proceso el valor saber jump
<i>Guardar 1(px)</i>	<i>s18</i>	8(s101)	JMP S100	Saltamos a cambiar de proceso
	<i>s19</i>	1(s1)	LDA s1	Llamemos k al proceso asociado a la dir px
	<i>s20</i>	4011	SHT 11	Se convierte 6(px) en 0(px)
	<i>s21</i>	2(c8)	ADD c8	Se convierte 0(px) en 1(px)
<i>Verificar si es el proceso 0</i>	<i>s22</i>	6(c0)	STO c0	Guarda en c0 1(px)
	<i>s23</i>	1c7	LDA c7	Se obtiene el id del proceso que se estaba ejecutando
	<i>s24</i>	7(000)	SUB 000	Se le resta 1 al id, acc=-1
<i>Si no lo es</i>	<i>s25</i>	3(s98)	BLZ s	Si acc<0 es el proceso 0 y se bloquea, no se borra
<i>Obtener 1-dir del siguiente proceso</i>	<i>s26</i>	1(c0)	LDA c0	Se obtiene 1(px)
	<i>s27</i>	2(c11)	ADD c11	Se obtiene 1(px)+5, la 1-dir del id del proceso siguiente (k+1)
	<i>s28</i>	6(s29)	STO s29	En s24 se guarda 1(px)+5, al que llamamos 1(py)
	<i>s29</i>	1(py)	LDA py	Se obtiene el id del proceso (k+1)
<i>¿Hay otro?</i>	<i>s30</i>	3(s52)	BLZ s52	Si acc<0 no hay otro proceso adelante
<i>Guardar 1(py) en c17</i>	<i>s31</i>	1(s29)	LDA s29	Se obtiene 1(py) en el acumulador
	<i>s32</i>	6(c17)	STO c17	Se guarda en c17 para usarla más tarde
<i>Inicializar contador de secciones</i>	<i>s33</i>	1(c18)	LDA 004	Se carga 0004 para funcionar como contador
	<i>s34</i>	6(c0)	STO C0	Se guarda el contador de secciones del proceso
<i>Verificar si faltan secciones</i>	<i>s35</i>	7000	SUB C11	Se verifica si ya termino con las secciones
	<i>s36</i>	3(s50)	BLZ s50	Si ya se pasaron las 4 secciones salta a ver si hay mas procesos
<i>Recorrer las secciones gpc, gacc, gjump, staticid de py a px</i>	<i>s37</i>	1(c17)	LDA c17	Se obtiene 1(py) en el acumulador
	<i>s38</i>	2(c0)	ADD Cc0	Obtiene la dirección del gpc/gacc/gjump/staticid del proceso py
	<i>s39</i>	6(s44)	STO s44	Se guarda en s44 para tener obtener el valor de la sección del proceso
	<i>s40</i>	7c11	SUB C11	Obtiene la dirección correspondiente del gpc(u otra sección) del proceso px
	<i>s41</i>	4011	SHT 11	Se convierte 1(px)+u en 0(px)+u
	<i>s42</i>	2(c9)	ADD c9	Se convierte 0(px)+u en 6(px)+u
	<i>s43</i>	6(s45)	STO s45	Se guarda en s45 para que sea cargada la sección en px
	<i>s44</i>	1(py)+u	LDA py+u	
	<i>s45</i>	6(px)+u	STO px+u	
	<i>s46</i>	1(c0)	LDA c0	Obtener contador de contextos
<i>Aumentar contador de secciones</i>	<i>s47</i>	7000	ADD 000	Se le resta uno al contador de contextos
	<i>s48</i>	6(c0)	STO c0	Lo guarda en c0 de nuevo
<i>Regresar a recorrer</i>	<i>s49</i>	8(s35)	JMP s35	
<i>¿Hay proceso después de (k+1)?</i>	<i>s50</i>	1(s29)	LDA S29	Se obtiene 1(py) en el acumulador
	<i>s51</i>	8(s27)	JMP s27	Va a verificar si hay otro proceso adelante
<i>"Si" de s30</i>	<i>s52</i>	1(s29)	LDA s29	Se carga la 1-dir del id del proceso k+1

Nombre clave	Dirección	Instrucción LM	Instrucción LE	Descripción
<i>Borrado</i>	s53	7(c11)	SUB c11	Se obtiene la 1-dir del id del proceso k, acc-=5
	s54	4011	SHT 11	Convierte 1(px) en 0(px)
	s55	2(c9)	ADD c9	Convierte 0(px) en 6(px)
<i>Borrado</i>	s56	6(s58)	STO s58	Guardar en s58 6(px)
	s57	1(c10)	LDA c10	Se cambia el valor del acumulador, acc=-1
<i>M[c4]--</i>	s58	6(px)	STO px	El proceso con id en la dir px ya no es accesible
	s59	1(c4)	LDA c4	Carga el id counter
	s60	7(000)	SUB 000	Se le resta 1 al id counter
<i>M[c5]=M[c5]-5</i>	s61	6(c4)	STO c4	Guardar en c4 el nuevo valor
	s62	1(c5)	LDA c5	Cargar el dir counter en el acumulador
	s63	7(c11)	SUB c11	Se le resta 4 al dir counter para estar a la par con el id counter
<i>M[c5]=M[c5]-5</i>	s64	6(c5)	STO c5	Se guarda en c5
	s65	8(s98)	JMP S97	Salta a ver si el id counter es el ultimo
<i>Aumenta el Id counter M[c4]++</i>	s66	1(s)	LDA s	Se carga en el acumulador la 8-dir de inicio del proceso
	s67	3(s95)	BLZ s94	si acc<0 no hay nuevo proceso
	s68	1(c4)	LDA c4	Se obtiene el valor del Id counter
<i>Aumenta el Dir counter M[c5] += 3</i>	s69	2000	ADD 000	
	s70	6(c4)	STO c4	
	s71	1(c5)	LDA c5	Se obtiene el valor del Dir counter
<i>Previa de ID</i>	s72	2(c11)	ADD c11	Se le suma 5 para tener la siguiente dirección
	s73	6(c5)	STO c5	
<i>Previa de pc</i>	s74	2(c9)	ADD c9	Convierte la nueva dirección en dir counter en 6-dir
	s75	6(s87)	STO s86	En s70 se guarda 6(px)
<i>Previa de acc</i>	s76	2000	ADD 000	Se obtiene la 6-dir del gpc del nuevo proceso
	s77	6(s89)	STO s78	En s72 se guarda 6(px)+1, la 6-dir del gpc del nuevo proceso
<i>Id del nuevo proceso</i>	s78	2000	ADD 000	Se obtiene la 6-dir del gacc del nuevo proceso
	s79	6(s92)	STO s81	En s75 se guarda 6(px)+2
<i>Guardar Nuevo Static ID para el proceso</i>	s80	2(c12)	ADD C12	Añade 2 para obtener 6(px)+4(Static ID)
	s81	6(s85)	STO s85	Guardar en s85
	s82	1(c15)	LDA c15	Obtener Serial de Static ID
	s83	2000	ADD 000	Añadir una unidad
	s84	6(c15)	STO C15	Guardar en c15 el folio actualizado
<i>pc nuevo</i>	s85	6(px)+4	STO (px)+4	Guarda el nuevo serial Static ID para el proceso
	s86	1(c4)	LDA c4	Se obtiene el Id para el nuevo proceso
<i>acc del nuevo proceso</i>	s87	6(px)	STO px	Se guarda el Id en la zona de proceso
	s88	1(s)	LDA s	Se obtiene el 8-pc del proceso
<i>Poner valor default en s</i>	s89	6(px)+1	STO px+1	Se guarda el 8-pc en el gpc del proceso
	s90	1000	LDA 000	
<i>¿está en el proceso 0?</i>	s91	7000	SUB 000	Se obtiene el 0
	s92	6(px)+2	STO px+2	El gacc del proceso se inicializa en 0
<i>¿Se acabaron los programas?</i>	s93	1(c10)	LDA c10	
	s94	6(s)	STO s	En s se coloca el valor -1
<i>Aumentar Id organizer M[c7]++</i>	s95	1(c7)	LDA c7	Se obtiene el id organizer
	s96	7(000)	SUB 000	Se le resta un 1, si es menor a 1 significa que es el p0
	s97	3(000)	BLZ 000	Si acc<0 salta al proceso 0
<i>Aumentar Dir organizer M[c6] += 5</i>	s98	1(c4)	LDA c4	Se obtiene el id counter, si es 1 hay que saltar a lanzar el proceso
	s99	7(000)	SUB 000	Se le resta un 1, acc=-1
	s100	3(s141)	BLZ s140	Si acc<0 salta al proceso 0
<i>Aumentar Dir organizer M[c6] += 5</i>	s101	1(c7)	LDA c7	
	s102	2000	ADD 000	
	s103	6(c7)	STO c7	Se aumenta el id organizer
<i>Aumentar Dir organizer M[c6] += 5</i>	s104	1(c6)	LDA c6	
	s105	2(c11)	ADD c11	
	s106	6(c6)	STO c6	Se aumenta el dir organizer

Nombre clave	Dirección	Instrucción LM	Instrucción LE	Descripción
<i>Verifica si hay que reiniciar</i>	<i>s107</i>	1(c4)	LDA c4	Se obtiene id counter
	<i>s108</i>	7(c7)	SUB c7	Al id counter se le resta el id organizer, M[c4]-M[c7]
	<i>s109</i>	3(s134)	BLZ s133	si acc<0 tenemos que reiniciar, por que llegamos al ultimo
<i>Sino-s109</i>	<i>s110</i>	1(c6)	LDA c6	Se obtiene la dir del id del proceso a ejecutar
<i>Actualizar s1</i>	<i>s111</i>	2(c9)	ADD c9	Se crea la 6-dir del id del proceso a ejecutar
	<i>s112</i>	6(s1)	STO s1	Se guarda en s1, será el proceso “actual”
	<i>s113</i>	1(c6)	LDA c6	
<i>Preparación para saltar al proceso con su pc y acc correctos</i>	<i>s114</i>	2(c8)	ADD c8	Se convierte en la 1-dir del id del proceso a ejecutar
	<i>s115</i>	2000	ADD 000	Se obtiene la 1-dir del gpc del proceso a ejecutar
<i>Continuación</i>	<i>s116</i>	6(s119)	STO s118	En s101 se guarda la 1-dir del gpc del proceso a ejecutar
	<i>s117</i>	2000	ADD 000	Se obtiene la 1-dir del gacc del proceso a ejecutar
	<i>s118</i>	6(s132)	STO s131	En 112 se guarda la 1-dir del gacc del proceso a ejecutar
<i>Preparación gpc y gcc</i>	<i>s119</i>	1(px)+1	LDA px+1	Obtiene el gpc del proceso a ejecutar
	<i>s120</i>	6(s133)	STO s132	Guarda el gpc del proceso a ejecutar en s106
	<i>s121</i>	1(s132)	LDA s131	Obtiene la 1 dir del gacc
<i>Preparación gjump</i>	<i>s122</i>	2000	ADD 000	Obtiene la 1 dir del gJump
	<i>s123</i>	6(s128)	STO s127	Guarda la 1 dir del gJump en s126
	<i>s124</i>	2000	ADD 000	Añadir un 1 para obtener el 1(px)+4, SID
<i>Salvar Static ID en 004</i>	<i>s125</i>	6(s126)	STO s125	Guardar en la siguiente celda
	<i>s126</i>	1(px)+4	LDA px+4	Obtener Static ID del proceso
	<i>s127</i>	6004	STO 004	Guardar en 0004 para usarlo como identificador el proceso que se está ejecutando
<i>Salvar gjump</i>	<i>s128</i>	1(px)+3	LDA px+3	Carga el valor de la gJump
	<i>s129</i>	6(c14)	STO c14	Guarda el gJump en c14 para que la arquitectura lo intercambie
	<i>s130</i>	1(000)	LDA 000	Obtiene el número 1
<i>bandera</i>	<i>s131</i>	6003	STO 003	Se permiten saltos con bandera==1
<i>Bandera</i>	<i>s132</i>	1(px)+2	LDA px+2	Se obtiene acc= acc del proceso a ejecutar
<i>LastDirectionSO</i>	<i>s133</i>	8(xx)	JMP xx	La dir xx representa el valor donde el proceso se ejecutará
<i>Reiniciar dir organizer</i>	<i>s134</i>	1(c3)	LDA c3	Se obtiene la primera dir de la zona de procesos, p3
	<i>s135</i>	6(c6)	STO c6	Se reinicia el dir organizer
	<i>s136</i>	2(c8)	ADD c8	Se crea la 1-dir de inicio en el acumulador
<i>Reiniciar id organizer</i>	<i>s137</i>	6(s138)	STO s137	En s138 se guarda px
	<i>s138</i>	1(px)	LDA px	Se obtiene el id de inicio
	<i>s139</i>	6(c7)	STO c7	Se reinicia el id organizer
<i>Regresar</i>	<i>s140</i>	8(s110)	JMP s110	Regresar para lanzar el proceso
<i>Reiniciar id organizer a 0</i>	<i>s141</i>	1(000)	LDA 000	
	<i>s142</i>	7000	SUB 000	Carga el id=0 del proceso 0
	<i>s143</i>	6(c7)	STO c7	Asignar a id organizer=0
<i>Asigna a 004 el 0</i>	<i>s144</i>	6(004)	STO 004	Guardar en 0004 el static Id 0 para el proceso 0
<i>Asigna a dir organizer=p0</i>	<i>s145</i>	1(c3)	LDA c3	Obtiene p4
	<i>s146</i>	7(c11)	SUB c11	Obtiene p1 al restarle 5
	<i>s147</i>	6(c6)	STO c6	Asignar a dir organizer=p0
	<i>s148</i>	8(000)	JMP 000	Saltar a proceso 0
Preámbulo				
Nombre clave	Dirección	Instrucción LM	Instrucción LE	Descripción
<i>Recibe el pc</i>	e0			Le coloca la maquina -1 si la instrucción fue 9(px), si salto el sw coloca el pc
<i>Manda acc a C</i>	e1	6(c2)	STO c2	Manda a c2 el último acc del proceso antes de saltar
<i>Cambiar bandera a “No” permitir</i>	e2	1(000)	LDA 000	Con bandera==0 no se permiten saltos
	e3	7(000)	SUB 000	No se permiten saltos
	e4	6(003)	STO 003	Saltamos a actualizar
<i>Mandar pc a C</i>	e5	1(e0)	LDA e0	Carga en el acumulador el último pc del proceso antes de saltar
	e6	2(c13)	ADD c13	Le coloca al pc el op-code 8
	e7	6(c1)	STO c1	En c1 se guarda el pc con op-code 8
<i>Obtener Saver Jump</i>	e8	1(999)	LDA 999	Cargar el último valor de 999 del proceso que salio
	e9	6(c14)	STO c14	Guardar en c14 el nuevo valor

Nombre clave	Dirección	Instrucción LM	Instrucción LE	Descripción
<i>Verificar marca</i>	e10	1(e0)	LDA e0	Se obtiene la marca
	e11	3(e13)	BLZ e11	Si la marca es menor a 1 se va al área de borrado
<i>Salto</i>	e12	8(s2)	JMP s2	Si la marca no es menor a 1 se va al área de actualización
<i>Salto</i>	e13	8(s19)	JMP s19	Salta al área de borrado
<i>Preámbulo para añadir procesos</i>	e14	1(e16)	LDA e16	Obtiene el numero máximo de procesos
	e15	7(c4)	SUB c4	Le resta la cantidad de procesos que hay
	e16	3(000)	BLZ 000	Si acc<0 se ha alcanzado el número máximo de procesos
	e17	8(s66)	JMP s66	Si no se ha alcanzado el máximo de procesos añadir otro
	e18			
Zona de Procesos				
Nombre clave	Dirección	Instrucción LM	Instrucción LE	Descripción
<i>Id Counter</i>	p0	0		Id del primer programa
<i>gpc</i>	p1	8000		Es el pc del proceso 0 con op-code 8
<i>gacc</i>	p2	0		Es el acumulador del proceso
<i>gjump</i>	p3	8000		Contiene el valor del que se tenía en la última dirección de memoria antes de saltar
<i>Static ID</i>	p4	0		Es el ID Static del proceso que no va a cambiar
<i>Id Counter</i>	p5	-1		Es el id del proceso correspondiente a esta sección
<i>gpc</i>	p6			Puede estar lleno de basura si no hay proceso en este contexto
<i>gacc</i>	p7			Puede estar lleno de basura si no hay proceso en este contexto
<i>gjump</i>	p8	8000		Por defecto tiene el 8000
<i>Static ID</i>	p9			
<i>Id Counter</i>	p10	-1		
<i>gpc</i>	p11			
<i>gacc</i>	p12			
<i>gjump</i>	p13	8000		
	p14			
	p15			
Variables del sistema				
Nombre clave	Dirección	Instrucción LM	Instrucción LE	Descripción
	c0			Espacio para el SOM
<i>8-pc</i>	c1			Espacio para el SOM/Preámbulo, recibe un valor en cada iteración
<i>acc</i>	c2			Espacio para el SOM/Preámbulo, recibe un valor en cada iteración
<i>Inicial</i>	c3	p5		Es la dir de inicio de la zona de procesos(usuario)
<i>Id counter</i>	c4	0		El contador de procesos de usuario
<i>Dir counter</i>	c5	p0		El contador de direcciones
<i>Dir organizer</i>	c6	p0		Contiene a la dirección del proceso que se ejecutará
<i>Id organizer</i>	c7	0		Contiene el id del proceso que se ejecutará
	c8	1000		Valor para convertir op-code en LOAD
	c9	6000		Valor para convertir op-code en STORE
	c10	-1		Valor de uso recurrente
	c11	5		Valor de uso recurrente, para el salto de los procesos
	c12	2		Valor de uso recurrente
	c13	8000		Valor para convertir op-code en JUMP
<i>Saver Jump</i>	c14			Guardará el valor de 999 por parte de la arquitectura
<i>Serial Id Procces</i>	c15	0		Contendrá un valor serial para los id de los procesos
<i>Máximo numero</i>	c16	4		Máximo numero de procesos disponibles (Menos 1 para la resta)
	c17	998		Espacio para el SOM/Área de borrado
	c18	4		Espacio para el SOM/Área de borrado