Proyecto de Arquitectura de Computadores Sistemas de Entrada/Salida

Jesús Vallejo Collados Sara Romano Rodríguez

ÍNDICE

Pseudocódigo y diagramas de flujo de cada subrutina	
Código comentado de cada subrutina	9
Pruebas usadas para cada subrutina	2
Observaciones y comentarios	27

Pseudocódigo y diagramas de flujo de cada subrutina

LEECAR

```
LEECAR(BUFFER(D0)){
     if (buffer.punteroEscritura != buffer.punteroLectura){
                     solucion= buffer.punteroLectura.lee();
                     if(buffer.punterolectura != buffer.finalBuffer){
                                  buffer.punteroLectura++;
              }
                      else {
                                    buffer.punteroLectura = buffer.iniciobuffer;
               }
     else {
                  return -1;
     return solucion;
}
       Leecar
                     PE -> PUNTERO ESCRITURA LINEA DESCRIPTOR
                    PL -> PUNTERO LECTURA LINEA DESCRIPTOR PB -> PUNTERO BUFFER LINEA DESCRIPTOR
                     FB -> FIN BUFFER LINEA DESCRIPTOR
    Comprobación
                                  D0 <- #-1
      PE == PL
         NO
     D0 <-(PL)+
                       NO
      PL == FB
                                    RTS
          SI
```

ESCCAR

```
ESCCAR(BUFFER(D0), CARACTER(D1)){
      buffer.punteroEscritura.Escribe(caracter);
      if(buffer.punteroEscritura == buffer.finalBuffer){
               buffer.punteroEscritura = buffer.inicioBuffer;
      if(buffer.punteroEscritura + 1 != buffer.punteroLectura){
               buffer.punteroEscritura++;
      else{
               return -1;
      buffer.punteroEscritura++;
      return 0;
}
                      PE -> PUNTERO ESCRITURA LINEA DESCRIPTOR
PL -> PUNTERO LECTURA LINEA DESCRIPTOR
PB -> PUNTERO BUFFER LINEA DESCRIPTOR
FB -> FIN BUFFER LINEA DESCRIPTOR
    Esccar
 Comprobación
   Descriptor
                             RTS
  D1 <- (PE)+1
D0 <- #0
                               NO
                                         SI
                 NO
                                                D0 <- #-1
   PE == FB
                           PE == PL
       SI
   PE <- PB
```

LINEA

```
LINEA(BUFFER(D0)){
      contador=0;
      A0 = buffer.punteroLectura;
      A1 = buffer.punteroEscritura;
      WHILE(A0!=A1){
                if(A0==A1){
                          return 0;
                contador++;
                 if(A0.leer() == 13){
                           return contador;
                Ã0++;
                if(A0 == buffer.finalBuffer){
                          A0=buffer.inicioBuffer;
      return 0;
}
                                      PE -> PUNTERO ESCRITURA LINEA DESCRIPTOR
                                      PL -> PUNTERO LECTURA LINEA DESCRIPTOR
PB -> PUNTERO BUFFER LINEA DESCRIPTOR
FB -> FIN BUFFER LINEA DESCRIPTOR
                    Comprobacion
Descriptor
                 D3 <- #0 (Contador=0)
                                       SI
                      PL==PE
                                               D0 <- #0
                         NO
              SI
  PL <- PB
                     PL == FB
                                                 RTS
                         NO
                                     SI
                    (PL) == #13
                                              D0 <- D3 + 1
                         NO
                    D3 <- D3 + 1
PL <- PL + 1
```

PRINT

```
PRINT (BUFFER, DESCRIPTOR, TAMAÑO(POR PILA)){
      if(descriptor == 0 || descriptor ==1){
               contador =0;
               while(contador!= tamaño && contadorCaracteres !=0){
                       caracter = Esscar(buffer.getCaracter());
                       CONTADOR++;
                       contadorCaracteres--;
                       if(caracter==13){
                              habilitaInterrupcionTransmision();
               dmpila();
      } else {
               return -1;
}
    CREACION MARCO DE PILA
                                                    DESTRUCCIÓN MARCO DE PILA
RTS
         TAMAÑO == 0
                              D0 <- #0
               NO
                      DISTINTO DE 0/1
        COMPROBACIÓN
DESCRIPTOR
                                         D0 <- #-1
    D2 <- #0 (CONTADOR=0)
                                                         D0 <- D2 (CONTADOR)
        D1 <- (BUFFER)+
                                 BSR ESCCAR
                                                   D0 == #-1
                                                       NO
               NO
          BSR ESCCAR
                                                  D2 <- D2 + 1
          D0 == #-1
                                          IMPEDIMOS INTERRUPCIONES
ACTIVO INTERRUPCION TRANSMISION
PERMITO INTERRUPCIONES
               NO
         D2 <- D2 + 1
                                       D2 == TAMAÑO
                                              ИО
```

```
SCAN (BUFFER, DESCRIPTOR, TAMAÑO(POR PILA)){
        if(descriptor == 0 || descriptor ==1){
                linea = linea();
                solucion = linea;
                if(linea==0){
                         return 0;
                }
                while(linea!=0){
                         buffer.punteroEscritura.escribir(leecar());
                         contadorCaracteres++;
                         if(buffer.punteroEscritura.getPos() == buffer.finalBuffer()){
                                 buffer.punteroEscritura= buffer.inicioBuffer;
                linea--;
                return solucion;
        } else {
        return -1;
}
       CREACIÓN MARCO DE PILA
                          DISTINTO DE 0/1
                                                                DESTRUCCIÓN MARCO DE PILA
RTS
           COMPROBACIÓN
DESCRIPTOR
                                             D0 <- #-1
             BSR LINEA
D2 <- D0
D3 <- D0
                                             D0 <- #0
             D2 == #0
                 NO
                               SI
           D2 > TAMAÑO
                 NO
                          SI
                                             D0 <- D3
             D2 == 0
           BSR LEECAR
(BUFFER)+ <- D0
D2 <- D2 - 1
```

```
RTI(){
       if(and(imrc,isr)==0 || and(imrc,isr)==2){
                 linea=linea();
                 if(retornoDeCarro==true){
                           buffer.escribirCaracter.transmision(10);
                           retornoDeCarro=false;
                           if(linea()==0){
                                     inhibirInterrupcionTransmision();
                           }
                 if(linea!=0 && linea!=-1){
                          caracter = leecar();
                          if(caracter == 13){
                                     retornoDeCarroTA=true;
                 buffer.escribirCaracter.transmision(caracter);
       if(and(imrc,isr)==1 || and(imrc,isr)==3){
                 esccar(buffer.getCaracter();
       }
}
                                                       RDCT = FLAG RETORNO DE CARRO
TB = TRANSMISSION BUFFER
RB = RECEPTION BUFFER
                          CREACION MARCO DE PILA
SALVAGUARDA DE REGISTROS COMPARTIDOS
D1 <- IMR & ISR
                                  COMPROBACIÓN
DESCRIPTOR D1
                                                  TRANSMISIÓN
                                       D1 == 1 EN BIT 0/1/4/5
      D1 <- RB
BSR ESCCAR
                                                                        RDCT == 1
                          RESTABLECIMIENTO REGISTROS COMPARTIDOS
DESTRUCCIÓN MARCO DE PILA
                                                                            NO
                                                                                               D0 == #1
                                                                       BSR LEECAR
                                                                                     INHIBICIÓN INTERRUPCIONES TRANSMISIÓN
                                                                                                 RDCT <- #1
                                                                        D0 == #13
                                                                            NO
                                                                        TB <- D0
```

Código comentado de cada subrutina

■ INIT()

INIT:

*Configuramos la Duart

	MOVE.B	#%00010000,CRA	* Reinicia el puntero MR1
	MOVE.B	#%00000011,MR1A	* 8 bits por caracter.
	MOVE.B	#%00010000,CRB	* Reinicia el puntero MR1
	MOVE.B	#%00000000,MR2A	* Eco desactivado.
	MOVE.B	#%00000011,MR1B	* 8 bits por caracter.
	MOVE.B	#%00000011, MR1B	* Eco desactivado.
	MOVE . D	##00000000, PIN2D	Leo desactivado:
	MOVE.B	#%11001100,CSRA	* Velocidad = 38400 bps.
	MOVE.B	#%11001100,CSRB	* Velocidad = 38400 bps.
	MOVE.B	#%00000000,ACR	* Velocidad = 38400 bps.
	MOVE.B	#%00000101 CDA	* Transmision v resension activades
	MOVE.D	#%00000101,CRA	* Transmision y recepcion activados.
	MOVE.B	#%00000101,CRB	* Transmision y recepcion activados.
	1101215	1170000001013 CRD	Transmission y recepción accivados.
	MOVE.B	#\$00000040,IVR	* Inicializa el vector a hex 40
	MOVE.B	#%00100010,IMR	* Inicializa interrupciones
escritu	ra	-,	
	MOVE.B	#%00100010,IMRC	* Inicializa interrupciones escritura
		•	·

```
MOVE.L
                  #BPA,A0
MOVE.L
                  A0, PPAE
MOVE.L
                  A0,PPAL
MOVE.L
                  #BPB,A0
                  A0,PPBE
MOVE.L
                  A0,PPBL
MOVE.L
MOVE.L
                  #BSA,A0
                  A0, PSAE
MOVE.L
                  A0, PSAL
MOVE.L
                  #BSB,A0
MOVE.L
                  A0, PSBE
MOVE.L
MOVE.L
                  A0,PSBL
```

*ACTUALIZAMOS LA DIRECCION DE LA TABLA DE VECTORES DE INTERRUPCION CON LA DIR DE LA RTI, cuando haya una interrupcion saltara a esa direccion

MOVE.L	#RTI,A0
MOVE.L	#\$100,A1
MOVE.L	A0,(A1)

RTS

■ LEECAR(BUFFER)

L	F	F	\mathbf{c}	Δ	R	•
_	_	_	┗.	_	ı١	

AND.L CMP.L BEQ CMP.L BEQ CMP.L BEQ CMP.L BEQ CMP.L	#3,D0 #0,D0 LESA #1,D0 LESB #2,D0 LEPA #3,D0 LEPB	*guardo los 2 bits mas significativos *comparo d0 con 00 *LEECAR SCAN A *comparo d0 con 01 *LEECAR SCAN B *comparo d0 con 10 *LEECAR PRINT A *comparo d0 con 11 *LEECAR PRINT B
	R SCAN A	
MOVE.L MOVE.L MOVE.L CMP.L *puntero de escritura	#0,D0 PSAE,A2 PSAL,A3 A2,A3 u lectura	*guardo puntero de escritura de A(scan) *guardo puntero de lectura de A(scan) *comparo los punteros-> Miro que los
*no esten a la misma a BEQ *D0=0xFFFFFFFF-> si es	FINLEC	*si son iguales devuelve en
*-1(buffer vacio) MOVE.B *devuelve el dato y se	(A3)+,D0	*devuelve en D0 el dato-> sino se
*puntero MOVE.L ADDA.L CMP.L *final del buffer se g	#2001,A4 #BSA,A4 A4,A3 uarda el puntero	*guardo 2001 en A4 *guardo la direccion final del buffer *comparo A4,A3-> si no se ha llegado al
BEQ MOVE.L *puntero	FLESA A3,PSAL	*guardo en la direccion el avance del
BRA FINLE		
FLESA: *FIN LEECA	R SCAN A	
MOVE.L *-> Si estamos al fina *pone el puntero al in	icio y se guarda	
MOVE.L *puntero BRA FINLE	A3,PSAL	*guardo en la direccion el avance del
LESB: *LEECAR SCA		
MOVE.L MOVE.L MOVE.L CMP.L BEQ	#0,D0 PSBE,A2 PSBL,A3 A2,A3 FINLEC	*guardo puntero de escritura de B(scan) *guardo puntero de lectura de B(scan) *comparo los punteros *si son iguales devuelve en
*D0=0xFFFFFFFF MOVE.B MOVE.L	(A3)+,D0 #2001,A4	*devuelve en D0 el dato *guardo 2001 en A4

	ADDA.L	#BSB,A4	*guardo la direccion final del buffer
	CMP.L	A4,A3	*comparo A4,A3
	BEQ	FLESB	w
*	MOVE.L	A3,PSBL	*guardo en la direccion el avance del
*puntero			
	BRA FINLE		
FLESB:	: *FIN LEECAR S	CAN D	
FLESD.	. FIN LLLCAN 3	CAN D	
	MOVE.L	#BSB,A3	*muevo el puntero a la direccion inicial
	MOVE.L	A3,PSBL	*guardo en la direccion el avance del
*puntero)	-	
	BRA FINLE		
LEPA:	_		
	MOVE.L	#0,D0	
	MOVE.L	PPAE,A2	*guardo puntero de escritura de
*A(print		DD41 43	*
	MOVE.L	PPAL,A3	*guardo puntero de lectura de A(print)
	CMP.L	A2,A3	*comparo los punteros
*D0 0vFF	BEQ	FINLEC	*si son iguales devuelve en
*D0=0xFF	MOVE.B	(A3)+,D0	*devuelve en D0 el dato
	MOVE.L	#2001,A4	*guardo 2001 en A4
	ADDA.L	#BPA,A4	*guardo la direccion final del buffer
	CMP.L	A4,A3	*comparo A4,A3 BEQ
FLEPA	CI II 1 L	A+,//3	comparo Atjas BEg
	MOVE.L	A3,PPAL	*guardo en la direccion el avance del
*puntero		,	
•			
	BRA FINLE		
FLEPA:		RINT A	
	: *FIN LEECAR P		*muevo el nunteno a la dinección inicial
	: *FIN LEECAR P	#BPA,A3	*muevo el puntero a la direccion inicial *guardo en la direccion el avance del
	: *FIN LEECAR P MOVE.L MOVE.L		*muevo el puntero a la direccion inicial *guardo en la direccion el avance del
	: *FIN LEECAR P MOVE.L MOVE.L	#BPA,A3	•
	: *FIN LEECAR P MOVE.L MOVE.L	#BPA,A3	·
	*FIN LEECAR P MOVE.L MOVE.L BRA FINLE	#BPA,A3 A3,PPAL	·
*punterc	: *FIN LEECAR P MOVE.L MOVE.L D BRA FINLE	#BPA,A3 A3,PPAL	·
*puntero	*FIN LEECAR P MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L	#BPA,A3 A3,PPAL B	·
*punterc	*FIN LEECAR P MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2	*guardo en la direccion el avance del *guardo puntero de escritura de
*puntero	*FIN LEECAR P MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L MOVE.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print)
*puntero	*FIN LEECAR P MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L MOVE.L CMP.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros
*puntero LEPB: *B(print	*FIN LEECAR P MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L MOVE.L CMP.L BEQ	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print)
*puntero LEPB: *B(print) *D0=0xFF	*FIN LEECAR P MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L E) MOVE.L CMP.L BEQ FFFFFFF	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en
*puntero LEPB: *B(print *D0=0xFF	*FIN LEECAR P MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L C) MOVE.L CMP.L BEQ FFFFFFF MOVE.B	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC (A3)+,D0	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en *devuelve en D0 el dato
*puntero LEPB: *B(print *D0=0xFF	MOVE.L MOVE.L D BRA FINLE *LEECAR PRINT MOVE.L MOVE.L C) MOVE.L CMP.L BEQ FFFFFFF MOVE.B MOVE.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC (A3)+,D0 #2001,A4	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en *devuelve en D0 el dato *guardo 2001 en A4
*puntero LEPB: *B(print *D0=0xFF	MOVE.L MOVE.L D BRA FINLE *LEECAR PRINT MOVE.L MOVE.L E) MOVE.L CMP.L BEQ FFFFFFF MOVE.B MOVE.L ADDA.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC (A3)+,D0 #2001,A4 #BPB,A4	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en *devuelve en D0 el dato *guardo 2001 en A4 *guardo la direccion final del buffer
*puntero LEPB: *B(print *D0=0xFF	MOVE.L MOVE.L D BRA FINLE *LEECAR PRINT MOVE.L MOVE.L E) MOVE.L CMP.L BEQ FFFFFFF MOVE.B MOVE.B MOVE.L ADDA.L CMP.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC (A3)+,D0 #2001,A4 #BPB,A4 A4,A3	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en *devuelve en D0 el dato *guardo 2001 en A4
*puntero LEPB: *B(print *D0=0xFF	MOVE.L MOVE.L MOVE.L D BRA FINLE *LEECAR PRINT MOVE.L MOVE.L CMP.L BEQ FFFFFFF MOVE.B MOVE.L ADDA.L CMP.L BEQ	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC (A3)+,D0 #2001,A4 #BPB,A4 A4,A3 FLEPB	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en *devuelve en D0 el dato *guardo 2001 en A4 *guardo la direccion final del buffer *comparo A4,A3
*puntero LEPB: *B(print *D0=0xFF	MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L CMP.L BEQ FFFFFFF MOVE.B MOVE.L ADDA.L CMP.L BEQ MOVE.L ADDA.L CMP.L BEQ MOVE.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC (A3)+,D0 #2001,A4 #BPB,A4 A4,A3	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en *devuelve en D0 el dato *guardo 2001 en A4 *guardo la direccion final del buffer
*puntero LEPB: *B(print *D0=0xFF	MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L CMP.L BEQ FFFFFFF MOVE.B MOVE.L ADDA.L CMP.L BEQ MOVE.L ADDA.L CMP.L BEQ MOVE.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC (A3)+,D0 #2001,A4 #BPB,A4 A4,A3 FLEPB	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en *devuelve en D0 el dato *guardo 2001 en A4 *guardo la direccion final del buffer *comparo A4,A3
*puntero LEPB: *B(print *D0=0xFF	MOVE.L MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L CMP.L BEQ FFFFFFF MOVE.B MOVE.L ADDA.L CMP.L BEQ MOVE.L BEQ MOVE.L BEQ MOVE.L BEQ MOVE.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC (A3)+,D0 #2001,A4 #BPB,A4 A4,A3 FLEPB A3,PPBL	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en *devuelve en D0 el dato *guardo 2001 en A4 *guardo la direccion final del buffer *comparo A4,A3
*puntero LEPB: *B(print *D0=0xFF	MOVE.L MOVE.L MOVE.L BRA FINLE *LEECAR PRINT MOVE.L MOVE.L CMP.L BEQ FFFFFFF MOVE.B MOVE.L ADDA.L CMP.L BEQ MOVE.L BEQ MOVE.L BEQ MOVE.L BEQ MOVE.L	#BPA,A3 A3,PPAL B #0,D0 PPBE,A2 PPBL,A3 A2,A3 FINLEC (A3)+,D0 #2001,A4 #BPB,A4 A4,A3 FLEPB A3,PPBL	*guardo en la direccion el avance del *guardo puntero de escritura de *guardo puntero de lectura de B(print) *comparo los punteros *si son iguales devuelve en *devuelve en D0 el dato *guardo 2001 en A4 *guardo la direccion final del buffer *comparo A4,A3

#BPB,A3 *muevo el puntero a la direccion inicial

MOVE.L

MOVE.L A3,PPBL *guardo en la direccion el avance del

*puntero

BRA FINLE

FINLEC: *FIN LEECAR CERO(NO HAY CARACTERES EN EL BUFFER)

MOVE.L #\$FFFFFFF,D0 *devuelve 0xFFFFFFF

FINLE: *FIN LEECAR

RTS

ESCCAR(BUFFER,CARÁCTER)

ESCCAR	•
--------	---

AND.W #3,D0 *me quedo con los dos bits

*significativos

CMP.W #0,D0

BEQ ESA *ESCCAR SCAN A

CMP.W #1,D0 BEQ ESB *ESCCAR SCAN B

CMP.W #2,D0

BEQ EPA *ESCCAR PRINT A

CMP.W #3,D0

BEQ EPB *ESCCAR PRINT B

ESA: *ESCCAR SCAN A

MOVE.L PSAE,A0 *CARGAMOS PUNTERO ESCRITURA MOVE.L PSAL,A2 *CARGAMOS PUNTERO LECTURA

MOVE.B D1,(A0)+ *ESCRIBIMOS EN BUFFER

MOVE.L #0,D0 *TODO BIEN

MOVE.L #BSA,A1

ADDA.L #2001,A1 *CARGO DIRECCION FINAL DEL

BUFFER

CMP.L A0,A1 *A0=A1??

BEQ ESAFB *FINAL DE BUFFER

CMP.L A0,A2 BNE FESA MOVE.L #-1,D0

BRA FINE

ESAFB: *ESCCAR SCAN A FIN DE BUFFER

MOVE.L #BSA,A0 *muevo el puntero a la

*direccion inicial

CMP.L A0,A2
BNE FESA
MOVE.L #-1,D0

BRA FINE

FESA: *FIN ESCCAR SCAN A

MOVE.L A0, PSAE

BRA FINE

EPA: *ESCCAR PRINT A

MOVE.L PPAE,A0 *CARGAMOS PUNTERO ESCRITURA MOVE.L PPAL,A2 *CARGAMOS PUNTERO LECTURA

MOVE.B D1,(A0)+ *ESCRIBIMOS EN BUFFER

MOVE.L #0,D0 *TODO BIEN

MOVE.L #BPA,A1

	ADDA.L	#2001,A1	*CARGO DIRECCION FINAL DEL
BUFFER	CMD I	AQ A1	*^^_^1)
	CMP.L BEQ	A0,A1 FPAFB	*A0=A1?? *FINAL DE BUFFER
	CMP.L	A0,A2	TINAL DE BOTTER
	BNE	FEPA	
	MOVE.L	#-1,D0	
	BRA FINE	•	
EPAFB:	*ESCCAR PRINT	A FIN DE BUFFER	
	MOVE.L	#BPA,A0	*muevo el puntero a la
*direccion			
	CMP.L	A0,A2	
	BNE	FEPA	
	MOVE.L BRA FINE	#-1,D0	
FEPA:	*FIN ESCCAR PF	ΣΤΝΤ Δ	
ILIA.	MOVE.L	A0,PPAE	
	BRA FINE	AOJITAL	
	2.0.1		
ESB:	*ESCCAR SCAN B		
	MOVE.L	PSBE,A0	*CARGAMOS PUNTERO ESCRITURA
	MOVE.L	PSBL,A2	*CARGAMOS PUNTERO LECTURA
	MOVE.B	D1,(A0)+	*ESCRIBIMOS EN BUFFER
	MOVE.L	#0,D0	*TODO BIEN
	MOVE.L	#BSB,A1	*CARCO RIRECCION EINAL REI
*BUFFER	ADDA.L	#2001,A1	*CARGO DIRECCION FINAL DEL
BUFFER	CMP.L	A0,A1	*A0=A1??
	BEQ	ESBFB	*FINAL DE BUFFER
	CMP.L	A0,A2	TIME DE BOTTEM
	BNE	FESB	
	MOVE.L	#-1,D0	
	BRA FINE		
ESBFB:		B FIN DE BUFFER	
	MOVE.L	#BSB,A0	*muevo el puntero a la
*direccion		40.42	
	CMP.L	A0,A2 FESB	
	BNE MOVE.L	#-1,D0	
	BRA FINE	π-1,00	
FESB:	*FIN ESCCAR SO	CAN B	
	MOVE.L	A0,PSBE	
	BRA FINE	•	
EPB:	*ESCCAR PRINT E	3	
	MOVE.L	PPBE,A0	*CARGAMOS PUNTERO ESCRITURA
	MOVE.L	PPBL,A2	*CARGAMOS PUNTERO LECTURA
	MOVE.B	D1,(A0)+	*ESCRIBIMOS EN BUFFER
	MOVE.L	#0,D0	*TODO BIEN
	MOVE.L ADDA.L	#BPB,A1 #2001,A1	*CARGO DIRECCION FINAL DEL
*BUFFER	ADDA • L	π2001,AI	CANGO DINECCION FINAL DEL
2011211	CMP.L	A0,A1	*A0=A1??
	BEQ	EPBFB	*FINAL DE BUFFER
	CMP.L	A0,A2	
	BNE	FEPB	
	MOVE.L	#-1,D0	
	BRA FINE		
EPBFB:	*ESCCAR PRINT	B FIN DE BUFFER	

```
MOVE.L
                            #BPB,A0
                                             *muevo el puntero a la
*direccion inicial
           CMP.L
                            A0,A2
                             FEPB
           BNE
           MOVE.L
                             #-1,D0
           BRA FINE
 FEPB:
             *FIN ESCCAR PRINT B
           MOVE.L
                            A0, PPBE
           BRA FINE
 FINE:
       RTS
         ■ LINEA(BUFFER)
 I TNFA:
       AND.W
                    #3,D0
                                        *Dos bits significativos
       CMP.L
                    #0,D0
       BEO
                    LSA
                                        *LINEA SCAN A
       CMP.W
                    #2,D0
       BEQ
                    LPA
                                        *LINEA PRINT A
       CMP.W
                    #1,D0
                                        *LINEA SCAN B
       BEQ
                    LSB
       CMP.W
                    #3,D0
       BEQ
                    LPB
                                         *LINEA PRINT B
 LSA:
          *LINEA SCAN A
       MOVE.L #0,D3
                                      *CONTADOR DE LINEA
       MOVE.L
                    #BSA,A4
       ADDA.L
                    #2001,A4
                                      *A0 PUNTERO CARGO PUNTERO DE LECTURA
       MOVE.L
                    PSAL,A0
                                      *CARGO PUNTERO DE ESCRITURA
       MOVE.L
                    PSAE,A1
 BLSA:
                                       *BUCLE LSA
       CMP.L
                    A1,A0
                                      *MIRO A VER SI ESTA FLV
                    FLV
       BEQ
                    A4,A0
       CMP.L
                    FLSA
       BEQ
       MOVE.B
                    (A0),D5
       CMP.L
                    #13,D5
                                     *USAMOS PUNTERO SCAN DE A LECTURA
                    FINLA
       BEO
       ADD.L
                    #1,D3
                                      *MUEVO EL PUNTERO
       ADDA.L
                    #1,A0
                    BLSA
       BRA
 FLSA: *FIN LINEA SCAN A
       MOVE.L
                    #BSA,A0
                                      *PONGO DIRECCION INICIAL EN EL PUNTERO
       BRA
                     BLSA
 LPA:
        *LINEA PRINT A
       MOVE.L
                    #0.D3
                                      *CONTADOR DE LINEA
       MOVE.L
                    #BPA,A4
       ADDA.L
                    #2001,A4
       MOVE.L
                    PPAL,A0
                                      *A0 PUNTERO
                                      *CARGO PUNTERO DE ESCRITURA
                    PPAE,A1
 BLPA: * BUCLE LINEA PRINT A
```

	CMP.L BEQ CMP.L BEQ	A1,A0 FLV A4,A0 FLPA	*MIRO A VER SI ESTA FLV
	MOVE.B CMP.L BEQ ADD.L	(A0),D5 #13,D5 FINLA #1,D3	*USAMOS PUNTERO PRINT DE A LECTURA
	ADDA.L BRA	#1,A0 BLPA	*MUEVO EL PUNTERO
FLPA:	*FIN LINEA MOVE.L BRA	PRINT A #BPA,A0 BLPA	*PONGO DIRECCION INICIAL EN EL PUNTERO
LSB:	*LINEA SCA	N B	
	MOVE.L MOVE.L ADDA.L	#0,D3 #BSB,A4 #2001,A4	*CONTADOR DE LINEA
	MOVE.L	PSBL,Ã0	*AØ PUNTERO
BLSB:	MOVE.L * BUCLE LI	PSBE,A1 NEA SCAN B	*CARGO PUNTERO DE ESCRITURA
	BEQ	A1,A0 FLV A4,A0 FLSB	*MIRO A VER SI ESTA FLV
	MOVE.B CMP.L BEQ ADD.L	(A0),D5 #13,D5 FINLA #1,D3	*USAMOS PUNTERO SCAN DE B LECTURA
	ADDA.L BRA	#1,A0 BLSB	*MUEVO EL PUNTERO
FLSB:	*FIN LINEA MOVE.L BRA	SCAN B #BSB,A0 BLSB	*PONGO DIRECCION INICIAL EN EL PUNTERO
LPB:	*LINEA PRI	NT B	
	MOVE.L	#0,D3 #BPB,A4	*CONTADOR DE LINEA
BLPB:	MOVE.L	#2001,A4 PPBL,A0 PPBE,A1 NEA PRINT B	*A0 PUNTERO LECTURA *CARGO PUNTERO DE ESCRITURA
	CMP.L BEQ CMP.L BEQ	A1,A0 FLV A4,A0 FLPB	*MIRO A VER SI ESTA FLV
	MOVE.B CMP.L BEQ ADD.L	#13,D5 FINLA #1,D3	*USAMOS PUNTERO PRINT DE B LECTURA
E. 55	ADDA.L BRA	#1,A0 BLPB	*MUEVO EL PUNTERO
FLPB:	*FIN LINEA MOVE.L BRA	#BPB,A0 BLPB	*PONGO DIRECCION INICIAL EN EL PUNTERO

FLV: *FIN LINEA VACIA

> MOVE.L #0,D0 *D0=0

RTS

FINLA: *FIN LINEA

ADD.L #1,D3 MOVE.L D3,D0

RTS

PRINT(BUFFER, DESCRIPTOR, TAMAÑO (POR PILA))

PRINT:

*CREAMOS MARCO DE PILA LINK A6,#0

*CARGAMOS DATOS

MOVE.L #0,A5 MOVE.L #0,D0 MOVE.L #0,D3

MOVE.L 8(A6),A5 *A5=buffer MOVE.W 12(A6),D0 *D0=descriptor MOVE.W *D1=TAMA? 14(A6),D3

#0,D3 CMP.L BEO **DMPILAT**

PRINTTL:

MOVE.L D0,D4 CMP.W #0,D0 *miro a ver en que puerto va a leer

BEQ PRINTA *escribe en puerto A

CMP.W #1.D0

PRINTB *escribe en puerto B BEO *D0=-1 SI NO ES NI 0 NI 1 MOVE.L #-1,D0 DMPILAPM

BRA

PRINTA: *PRINT PUERTO A

MOVE.L #0,D2 *D2=CONTADOR

BUCPA:

MOVE.L #2,D0 *PREPARO DO PARA ESCCAR

MOVE.L *LIMPIO D1 #0,D1

MOVE.B (A5)+,D1*OBTENGO EL CARACTER DEL BUFFER

CMP.L #13,D1 *MIRO A VER SI ES RETORNO DE CARRO

BEQ ACTTA

BUCPA1:

ESCCAR *LLAMO A ESCCAR, escribo el caracter BSR #-1,D0 *COMPRUEBO VALOR DEVUELTO POR ESCCAR CMP.L

DMPILAP BEO

ADD.L #1.D2 *CONTADOR++

CMP.L D2,D3 *MIRO A VER SI HEMOS LLEGADO HASTA

TAMAÑO

BEQ DMPILAP

İ	BRA	BUCPA		*SIGUIENTE CARACTER
ACTTA:				
I	BSR	ESCCAR		*LLAMO A ESCCAR
	CMP.L	#-1,D0		*COMPRUEBO VALOR DEVUELTO POR ESCCAR
I	BEQ	DMPILAP		
	ADD.L	#1,D2		*CONTADOR++
DMPILA	PA:			
2	MOVE.W	#\$2700,SR		*ACTIVO TRANSMISION
	BSET	#0,IMRC		
	MOVE.B	IMRC, IMR		
	MOVE.W	#\$2000,SR		
	BRA	DMPILAP		
PRINTB	: *PRINT PUER	то в		
ı	MOVE.L	#0,D2		*D2=CONTADOR
BUCPB:		,		
I	MOVE.L	#3,D0		*PREPARO DØ PARA ESCCAR
MOVE.L	#0,D1		*LIMPIO	D1
1	MOVE.B	(A5)+,D1		*OBTENGO EL CARACTER DEL BUFFER
(CMP.L	#13,D1		*MIRO A VER SI ES RETORNO DE CARRO
	BEQ	ACTTB		
BUCPB1				
	BSR	ESCCAR		*LLAMO A ESCCAR
	CMP.L	#-1,D0		*COMPRUEBO VALOR DEVUELTO POR ESCCAR
	BEQ	DMPILAP		d-0017-17-07
	ADD.L	#1,D2		*CONTADOR++
	CMP.L	D2,D3		*MIRO A VER SI HEMOS LLEGADO HASTA
*TAMAÑO	DEO	DMPILAP		
	BEQ BRA	BUCPB		
ACTTB:	DNA	ВОСРВ		
	BSR	ESCCAR		*LLAMO A ESCCAR
	CMP.L	#-1,D0		*COMPRUEBO VALOR DEVUELTO POR ESCCAR
	BEQ	DMPILAP		CONTROLLO VALOR BEVOLETO FOR ESCONIK
	ADD.L	#1,D2		*CONTADOR++
DMPILA		"#2700 CD		*ACTIVO TRANSMICTON
	MOVE.W	#\$2700,SR		*ACTIVO TRANSMISION
	BSET MOVE B	#4,IMRC		
	MOVE.B MOVE.W	IMRC,IMR #\$2000,SR		
	BRA	##Z000,3K		
	DIA	DITTLAP		
DMPILA		#0,D2		
DMPILA		D2,D0		*METO EL NUMERO DE CARACTERES ESCRITOS
DMPILA				
	UNLK	A6		*DESTRUCCION DE MARCO DE PILA
	RTS			

■ SCAN(BUFFER,DESCRIPTOR,TAMAÑO(POR PILA))

SCAN:

LINK A6,#0 *CREAMOS MARCO DE PILA

MOVE.L #0,D0

	MOVE.L	#0,D1	
	MOVE.W	14(A6),D1	*D1=tamaño
	MOVE.W	12(A6),D0	*D0=descriptor
	MOVE.L	8(A6),A0	*A0=buffer CARGO EL Buffer
	CMP.W	#0,D0	*miro a ver en que puerto va a leer
	BEQ	SCANA	*escribe en puerto A
	CMP.W	#1,D0	
	BEQ	SCANB	*escribe en puerto B
	MOVE.L	#-1,D0	*D0=-1 SI NO ES NI 0 NI 1
	BRA	DMPÍLAS	
SCANA:	*SCAN PUERTO	А	
	MOVE.B	#0,D0	*D0=0 BUSCAMOS CUANTOS CARACTERES
*HAY FN	EL BUFFER DE S	-	
	BSR	LINEA	
	MOVE.L	8(A6),A0	*A0=buffer CARGO EL Buffer
	MOVE.L	D0,D2	*D2=I TNFA
	CMP.W	#0,D2	*LINEA=0?
	BEQ	FINCEROA	LINEA-U.
	MOVE.L	D2,D3	*D3 REGISTRO TAMAÑO ESCRITO
	CMP.W	D1,D2	*COMPARO TAMA? Y LINEA
	BGT	FINCEROA	COMPARO TAMA: I LINEA
BUCSA:		FINCEROA	
BUCSA:	CMP.W	#0 D2	*LINEA=0?
		#0,D2	"TINEA=A:
	BEQ	FINSCANA	*DADAMETRO DADA LEECAR
	MOVE.B	#0,D0	*PARAMETRO PARA LEECAR
	BSR	LEECAR	
	MOVE.B	D0,(A0)+	*COPIO EL CARACTER EN BUFFER
	SUB.L	#1,D2	*N
	BRA BUCSA		
FINCER	OA: MOVE.L	#0,D0	*DEVUELVE 0 EN D0
	BRA DMPILAS		
FINSCA	NA:		
	MOVE.L	D3,D0	*D0=N
	BRA DMPILAS		
SCANB:	*SCAN PUERTO		
	MOVE.L	#0,D0	
	MOVE.L	#1,D0	*D0=0
	BSR	LINEA	*LLAMO A LINEA PARA SABER CUAL ES
*EL TAMA	ÑO DE LINEA		
	MOVE.L	8(A6),A0	*A0=buffer CARGO EL Buffer
	MOVE.L	DÔ, D2	*D2=LINEA
	CMP.W	#0,D2	*LINEA=0?
	BEQ	FINCEROB	
	MOVE.L	D2,D3	*D3 REGISTRO TAMAÑO ESCRITO
	CMP.W	D1,D2	*COMPARO TAMA? Y LINEA
	BGT	FINCEROB	COLUMN TO THE PERSON OF THE PE
BUCSB:	-	. INCLINOD	
י מכיסה	CMP.W	#0,D2	*LINEA=0?
	BEQ	FINSCANB	LINLA-U;
	MOVE.L	#1,D0	*PARAMETRO PARA LEECAR
	BSR	#I,D0 LEECAR	TANAPILINO FANA LEECAN
			*COPIO EL CARACTER EN BUFFER
	MOVE.B	D0,(A0)+	
	SUB.L	#1,D2	*N
	BRA BUCSB		

```
BRA DMPILAS
  FINSCANB: MOVE.L
                       D3,D0
                                              *D0=N
          BRA DMPILAS
  DMPILAS:
                                            *DESTRUCCION MARCO DE PILA
        UNLK
                     Α6
        RTS
           RTI()
  RTI:
        LINK
                       A6,#-36
                                            *Guardamos todos los registros para
*asegurar que no hay problemas de
*concurrencia
        MOVE.L
                       D0,-4(A6)
                       D1,-8(A6)
        MOVE.L
        MOVE.L
                       D3,-12(A6)
                       D5,-16(A6)
        MOVE.L
        MOVE.L
                       A0,-20(A6)
                       A1,-24(A6)
        MOVE.L
                       A2,-28(A6)
        MOVE.L
        MOVE.L
                       A3,-32(A6)
        MOVE.L
                      A4,-36(A6)
 AMBOS:
        MOVE.L
                      #0,D1
        MOVE.B
                      IMRC,D1
                                          *COPIO EN UN REGISTRO LA COPIA DEL
*IMR
                    ISR,D1
        AND.B
                                        *FUNCION AND entre iSR y imrc
             #0,D1
                                  *MIRO EL BIT 0 DE D1
BTST
        BNE
                        TΑ
                      #1,D1
                                            *MIRO EL BIT 1 DE D1
        BTST
        BNE
                        RA
                                            *MIRO EL BIT 4 DE D1
        BTST
                      #4,D1
        BNE
                        TB
                                           *MIRO EL BIT 5 DE D1
        BTST
                      #5,D1
        BNE
                        RB
                    FINREAL
        BRA
 TA: *TRANSMISION DE A
            CMP.L
                         #1,RDCTA
                         FINTA
            BEQ
            MOVE.L
                         #2,D0
                                            *METO EN D0 3
            BSR
                         LEECAR
                                             *RETORNO DE CARRO?
            CMP.L
                         #13,D0
                         VUELTATA
            BNE
  RETCATA:
            MOVE.L
                         #1,RDCTA
  VUELTATA:
          MOVE.B
                         D0,TBA
          BRA
                         FINRTI
  FINTA:
```

*DEVUELVE 0 EN D0

FINCEROB: MOVE.L

MOVE.L

#0,RDCTA

#0,D0

MOVE.B	#10,TBA	*NO MAS LINEA, SI RET DE CARRO,
*METO SALTO DE LINEA	,	, , , , , , , , , , , , , , , , , , , ,
MOVE.L	#2,D0	
BSR	LINEA	
CMP.L	#0,D0	*LINEA =0?
BNE	FINRTI	
BCLR	#0,IMRC	*INHIBO INTERRUPCIONES EN TA
MOVE.B	IMRC, IMR	
BRA	FINRTI	
TB: *TRANSMISION DE B		
CMP.L	#1,RDCTB	
BEQ	FINTB	
MOVE.L	#3,D0	*METO EN DØ 3
BSR	LEECAR	
CMP.L	#13,D0	*RETORNO DE CARRO?
BNE	VUELTATB	
RETCATB:		
MOVE.L	#1,RDCTB	
VUELTATB:		
MOVE.B	D0,TBB	*NO DETORNO DE CARRO ET LINEA
*METO CARACTER	ססו, שט	*NO RETORNO DE CARRO, SI LINEA,
BRA	FINRTI	
FINTB:	LINKIT	
	#A DDCTD	
MOVE.L	#0,RDCTB	*NO MAC LINEAC CI DET DE CADDO
MOVE.B	#10,TBB	*NO MAS LINEAS, SI RET DE CARRO,
*METO SALTO DE LINEA	#2 D0	
MOVE.L	#3,D0	
BSR	LINEA	VI TNEA OO
CMP.L	#0,D0	*LINEA =0?
BNE	FINRTI	
BCLR	#4,IMRC	*INHIBO INTERRUPCIONES EN TB
MOVE.B	IMRC, IMR	INITIDO INTERROL CIONES EN TO
BRA	FINRTI	
DIVA	TIMILIT	
RA: *RECEPCION DE A		
MOVE.L	#0,D1	
MOVE.B	RBA,D1	*CARACTER PARA ESCCAR
MOVE.L	#0,D0	*BUFFER PARA ESCCAR(RBA)
BSR	ESCCAR	
BRA	FINRTI	
RB: *RECEPCION DE B		
	#A D1	
MOVE.L MOVE.B	#0,D1	*CARACTER PARA ESCCAR
MOVE.B MOVE.L	RBB,D1 #1,D0	
BSR	#1,00 ESCCAR	*BUFFER PARA ESCCAR(RBB)
BRA	FINRTI	
DNA	LTINUIT	
FINRTI:		
BRA AMBOS		*COMPRUEBA SI HAY MÁS
*TNTERRIDCTONES		

*INTERRUPCIONES

```
FINREAL:
     MOVE.L
                     -36(A6),A4
                                               *DESTRUCCION MARCO DE PILA
     MOVE.L
                     -32(A6),A3
     MOVE.L
                     -28(A6),A2
                     -24(A6),A1
     MOVE.L
                     -20(A6),A0
     MOVE.L
                     -16(A6),D5
     MOVE.L
                     -12(A6),D3
     MOVE.L
     MOVE.L
                     -8(A6),D1
     MOVE.L
                     -4(A6),D0
     UNLK
                      Α6
     RTE
```

Listado de pruebas para las subrutinas

Hemos realizado muchas más pruebas, siguiendo la estructura de las mostradas en este apartado y cambiando los datos de las mismas. A continuación se muestra un ejemplo de las pruebas usadas en cada subrutina:

Prueba LEECAR

INICIO:

BSR	INIT
MOVE.L	#0,A5
MOVE.L MOVE.B MOVE.L MOVE.N BSR MOVE.W BSR	PSAE,A5 #31,(A5)+ A5,PSAE PSBE,A5 #32,(A5)+ A5,PSBE PPAE,A5 #33,(A5)+ A5,PPAE PPBE,A5 #34,(A5)+ A5,PPBE #0,D0 LEECAR #1,D0 LEECAR
MOVE.W	#2,D0
BSR	LEECAR
MOVE.W	#3,D0
BSR BRFAK	LEECAR
DKEAK	

Prueba ESCCAR

INICIO:

ABUCLEE:

MOVE.L #1500,D7

```
ADD.L
                   #1,D6
       MOVE.L
                   D6,D1
  BUCLEE:
      \mathsf{CMP.L}
                  #0,D7
      BEQ
                  ABUCLEL
                  #2,D0
      MOVE.L
      BSR
                  ESCCAR
      SUB.L
                  #1,D7
      BRA
                  BUCLEE
  ABUCLEL:
       MOVE.L
                    #1500,D7
  BUCLEL:
                  #0,D7
      CMP.L
      BEQ
                  ABUCLEE2
      MOVE.L
                  #2,D0
      BSR
                  LEECAR
                  #1,D7
      SUB.L
      BRA
                  BUCLEL
ABUCLEE2:
    MOVE.L
                 #999,D7
    BREAK
                #1,D6
    ADD.L
    {\tt MOVE.L}
                D6,D1
  BUCLEE2:
     CMP.L
                 #0,D7
     BEQ
                 ABUCLELI
     MOVE.L
                 #2,D0
                 ESCCAR
     BSR
     SUB.L
                 #1,D7
                 BUCLEE2
     BRA
  ABUCLELI:
     MOVE.L
                 #13,D1
     MOVE.L
                 #2,D0
                  ESCCAR
      BSR
     MOVE.L
                #2,D0
     BREAK
     BSR
                LINEA
     BREAK
     FIN:
              BREAK
     BUS_ERROR:BREAK
                                          * Bus error handler
                NOP
                                           * Address error handler
     ADDRESS_ER:BREAK
                  NOP
     ILLEGAL_IN:BREAK
                                           * Illegal instruction handler
                                           * Privilege violation handler
     PRIV_VIOLT: BREAK
          Prueba LINEA
 INICIO:
                  INIT
     BSR
     MOVE.W
                        #1,D0
```

MOVE.W

#1,D1

BSR	ESCCAR
MOVE.W	#1,D0
BSR	LEECAR
MOVE.W	#1,D0
MOVE.W	#3,D1
BSR	ESCCAR
MOVE.W	#1,D0
MOVE.W	#4,D1
BSR	ESCCAR
MOVE.W	#1,D0
MOVE.W	#13,D1
BSR	ESCCAR
MOVE.L	#1,D0
BSR	LINEA

BREAK

Prueba PRINT

• Prueba 6 caracteres

BUFP:	DS.B	2100	*Buffer para lectura y escritura de
*caracteres CONTLP: CONTCP: DIRLECP: DIRESCP: TAMEP: DESAP: DESBP: NLINP: TAMLP: TAMBP:	DC.W DC.L DC.L DC.W EQU EQU EQU EQU EQU EQU	0 0 0	*Contador de lineas *Contador de caracteres *Direccion de lectura para SCAN *Direccion de escritura para PRINT *Tamaño de escritura para PRINT *Descriptor de linea A *Descriptor de linea B *Numero de lineas a leer *Tamaño de linea para SCAN *Tamaño de bloque para PRINT
INICIO:			
	*MOVE.L *MOVE.L *MOVE.L *MOVE.L *MOVE.W MOVE.B	#BUS_ERROR #ADDRESS_E #ILLEGAL_I #PRIV_VIOL INIT #\$2000,SR #BUFP,A0 #1,(A0)+ #2,(A0)+ #3,(A0)+ #4,(A0)+ #5,(A0)+ #13,(A0) #BUFP,A0 #0,D0 #1,D3 D3,-(A7) D0,-(A7)	R,12 * Address error handler N,16 * Illegal instruction handler T,32 * Privilege violation handler *Permite interrupciones
	MOVE.L BSR	A0,-(A7) PRINT	

• Prueba 1900 caracteres

BUFP:	DS.B	2100	*Buffer para lectura y escritura de
caracteres CONTLP:	DC.W	0	*Contador de lineas
CONTEP:	DC.W	0	*Contador de caracteres
DIRLECP:	DC.L	0	*Direccion de lectura para SCAN
DIRESCP:	DC.L	0	*Direccion de escritura para PRINT
TAMEP:	DC.W	0	*Tamaño de escritura para PRINT
DESAP:	EQU	0	*Descriptor de linea A
DESBP:	EQU	1	*Descriptor de linea B
NLINP:	EQU	1	*Numero de lineas a leer
TAMLP:	EQU	30	*Tamaño de linea para SCAN
TAMBP:	EQU	30	*Tamaño de bloque para PRINT
TNTCTO			
INICIO:			
	BSR	INIT	
	MOVE.W	#\$2000,	•
	MOVE.L	#BUFP,A	
	MOVE.L	#190,D2	*contador
	MOVE.L	#0,D4	
BUC:	CMD I	D4 D2	
	CMP.L	D4,D2	
	BEQ MOVE B	FIN	
	MOVE B	#1,(A0)+	
	MOVE.B MOVE.B	#2,(A0)+ #3,(A0)+	
	MOVE.B	#4,(A0)+	
	MOVE.B	#5,(A0)+	
	MOVE.B	#6,(A0)+	
	MOVE.B	#7,(A0)+	
	MOVE.B	#8,(A0)+	
	MOVE.B	#9,(A0)+	
	MOVE.B	#0,(A0)+	
	ADD.L	#1,D4	
	BRA	BUC	
FIN:			
	MOVE.B	#13,(A0)	
	MOVE.L	#BUFP,A	0
	MOVE.W	#1,D0	_
	MOVE.W	#1901,D	
	MOVE.W	D3,-(A7	
	MOVE.W	D0,-(A7	
	MOVE.L	A0,-(A7)
	BSR	PRINT	
	BREA	K	

■ Prueba SCAN

BUFP:	DS.B	2100	*Buffer para lectura y escritura de
caracteres			
CONTLP:	DC.W	0	*Contador de lineas
CONTCP:	DC.W	0	*Contador de caracteres
DIRLECP:	DC.L	0	*Direccion de lectura para SCAN
DIRESCP:	DC.L	0	*Direccion de escritura para PRINT
TAMEP:	DC.W	0	*Tamaño de escritura para PRINT
DESAP:	EQU	0	*Descriptor de linea A
DESBP:	EQU	1	*Descriptor de linea B
NLINP:	EQU	1	*Numero de lineas a leer
TAMLP:	EQU	30	*Tamaño de linea para SCAN
TAMBP:	EQU	30	*Tamaño de bloque para PRINT

INICIO:

BSR	INIT	
MOVE.W MOVE.W BSR MOVE.W BSR MOVE.W BSR MOVE.W MOVE.W MOVE.W MOVE.W MOVE.L MOVE.L MOVE.W MOVE.W MOVE.W	#\$2000,SR *Permite interrupcione #0,D0 #1,D1 ESCCAR #0,D0 #2,D1 ESCCAR #0,D0 #13,D1 ESCCAR #0,A0 #BUFP,A0 #0,D0 #3,D3 D3,-(A7)	÷S
MOVE.L BSR BREAK	A0,-(A7) * 32E4 SCAN * SEE 2A60	

Prueba RTI

BUFFER:	DS.B	2100	*Buffer para lectura y escritura
*de caracto	eres		
CONTL:	DC.W	0	*Contador de lineas
CONTC:	DC.W	0	*Contador de caracteres
DIRLEC:	DC.L	0	*Direccion de lectura para SCAN
DIRESC:	DC.L	0	*Direccion de escritura para
PRINT			
TAME:	DC.W	0	*Tamaño de escritura para PRINT
DESA:	EQU	0	*Descriptor de linea A
DESB:	EQU	1	*Descriptor de linea B
NLIN:	EQU	1	*Numero de lineas a leer

TAML:	EQU	30	*Tamaño de linea para SCAN
TAMB:	EQU	5	*Tamaño de bloque para PRINT
INICIO:	MOVE	WDUG EDDOD O	* B
	MOVE.L	#BUS_ERROR,8	* Bus error handler
	MOVE.L	#ADDRESS_ER,12	* Address error handler
	MOVE.L	#ILLEGAL_IN,16	* Illegal instruction handler
	MOVE.L BSR	#PRIV_VIOLT,32 INIT	<pre>* Privilege violation handler</pre>
	MOVE.W	#\$2000,SR	*Permite interrupciones
	MOVE.W	#\$2000,3K	remitte interrupciones
BUCPR:	MOVE.W	#0,CONTC	*Inicializa contador de caracteres
	MOVE.W	#NLIN,CONTL	*Inicializa contador de lineas
	MOVE.L	#BUFFÉR,DIRLEC	*Direccion de lectura (comienzo
del buffer)	•	· ·
OTRAL:	MOVE.W	#TAML,-(A7)	*Tamaño maximo de la linea
	MOVE.W	#DESA, - (A7)	*Puerto A
	MOVE.L	DIRLEĆ,-(A7)	*Direccion de lectura
ESPL:	BSR	SCAN	
	CMP.L	#0,D0	
	BEQ	ESPL	*Si no se ha leido una linea de
*intenta d	e nuevo		
	ADDA.L	#8,A7	*Restablece la pila
	ADD.L	D0,DIRLEC	*Calcula la nueva direccion de
*lectura			
	ADD.W	D0,CONTC	*Actualiza el numero de caracteres
*leidos			
	SUB.W	#1,CONTL	*Actualiza el numero de lineas
*leidas. S	-		
	BNE	OTRAL	*se han leido todas las lineas se
*vuelve a	leer		
	MOVE.L	#BUFFER,DIRLEC	*Direccion de lectura (comienzo
*del buffe	r)	•	`
OTRAE:	MOVE.W	#TAMB,TAME	*Tamaño de escritura = Tamaño de
*bloque		•	
ESPE:	MOVE.W	TAME,-(A7)	*Tamaño de escritura
	MOVE.W	#DESA,-(A7)	*Puerto A
	MOVE.L	DIRLEC,-(A7)	* Direccion de lectura
	BSR	PRINT	
	ADD.L	#8,A7	*Restablece la pila
	ADD.L	D0,DIRLEC	*Calcula la nueva direccion del
*buffer			
	SUB.W	D0,CONTC	*Actualiza el contador de caracteres
	BEQ	SALIR	*Si no quedas caracteres se acaba
	SUB.W	D0,TAME	*Actualiza el tamaño de escritura
	BNE	ESPE	*Si no se ha escrito todo el bloque
*se insist	_	UTAME - 001155	*C. 1 0 1
de	CMP.W	#TAMB,CONTC	*Si el nº de caracteres que quedan
*es menor	dne eī		*tamaño octablacida co transimita
*ese numer	0		*tamaño establecido se transimite
ese numer	o BHI	OTRAE	*Siguiente bloque
	MOVE.W	CONTC, TAME	organice produc
	BRA	ESPE	*Siguiente bloque
	DIA	LJIL	Significe broduc
SALIR:	BRA	BUCPR	
J ±			

FIN:

BREAK

BUS_ERROR:

BREAK

NOP

ADDRESS_ER: BREAK NOP

ILLEGAL_IN:

BREAK

NOP

PRIV_VIOLT:

BREAK NOP

Observaciones y comentarios