

Proyecto de Arquitectura de Computadores
Sistemas de Entrada/Salida

Jesús Vallejo Collados
Sara Romano Rodríguez

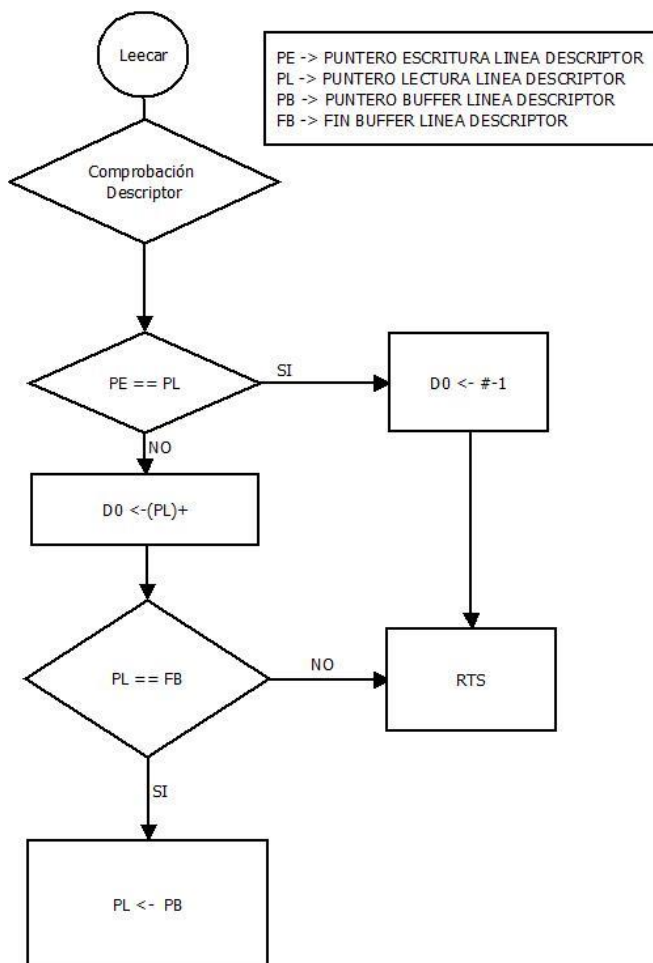
ÍNDICE

Pseudocódigo y diagramas de flujo de cada subrutina	3
Código comentado de cada subrutina	9
Pruebas usadas para cada subrutina	21
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;  
}
```

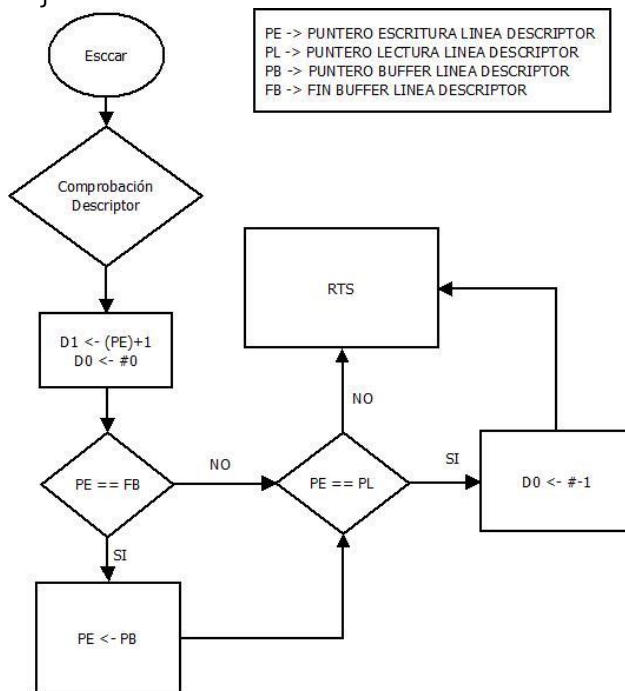


- ESCCAR

```

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;
}

```

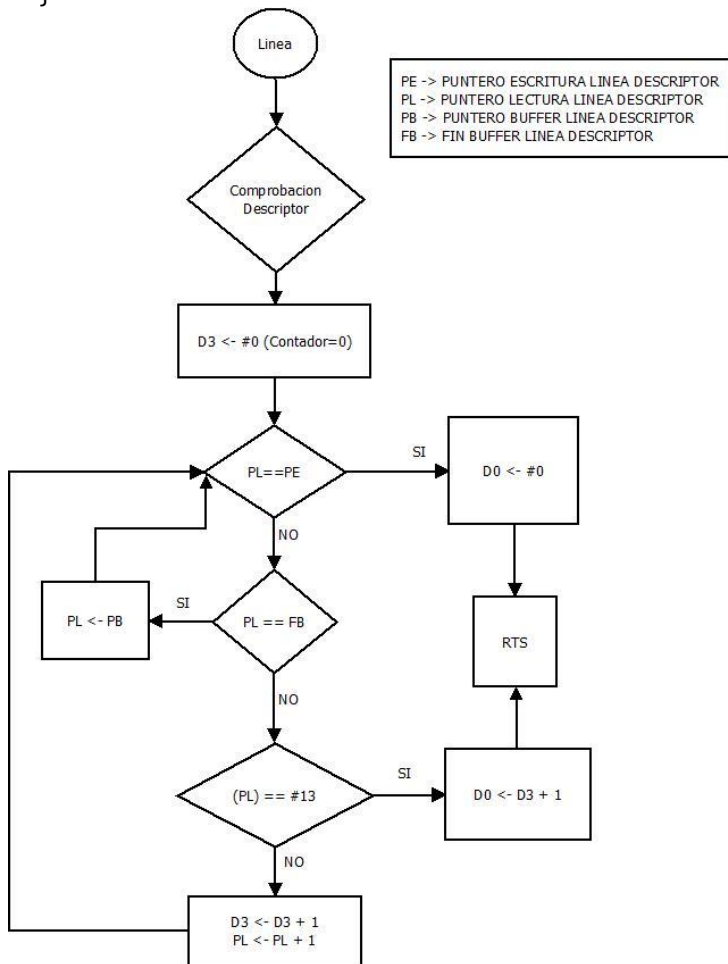


▪ LINEA

```

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;
        }
        A0++;
        if(A0 == buffer.finalBuffer){
            A0=buffer.inicioBuffer;
        }
    }
    return 0;
}

```

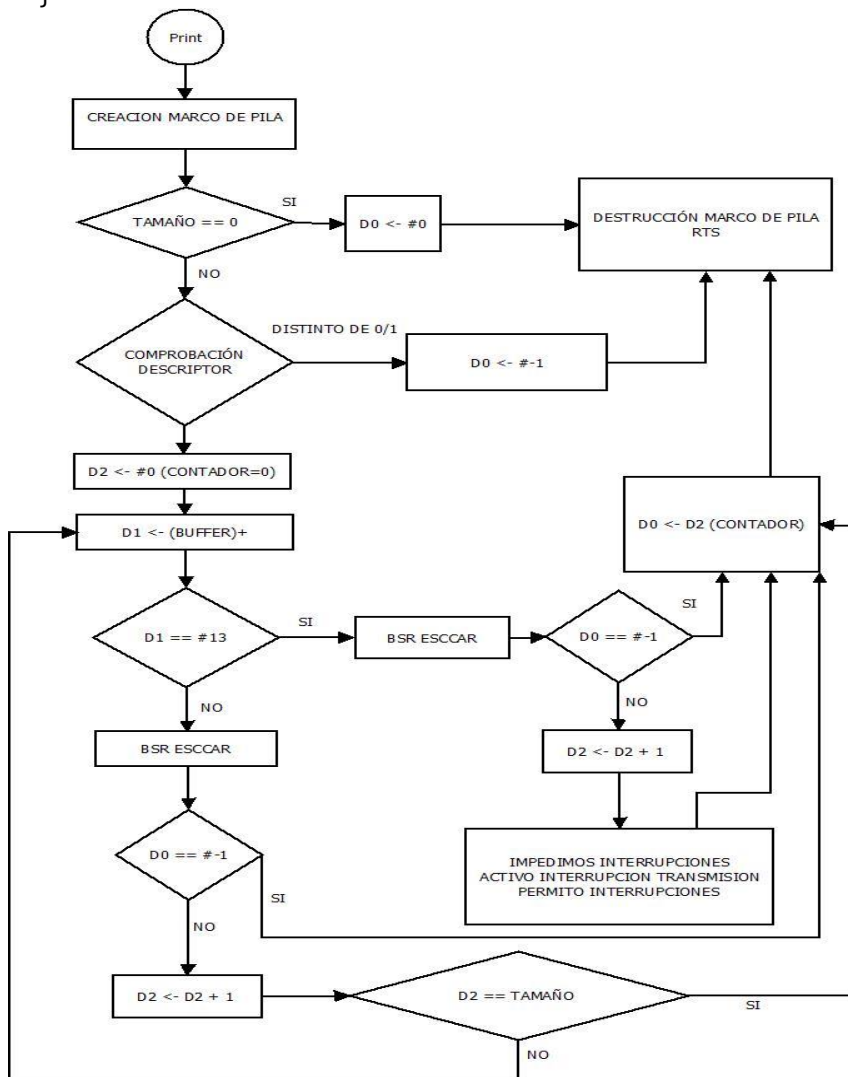


- PRINT

```

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;
    }
}

```

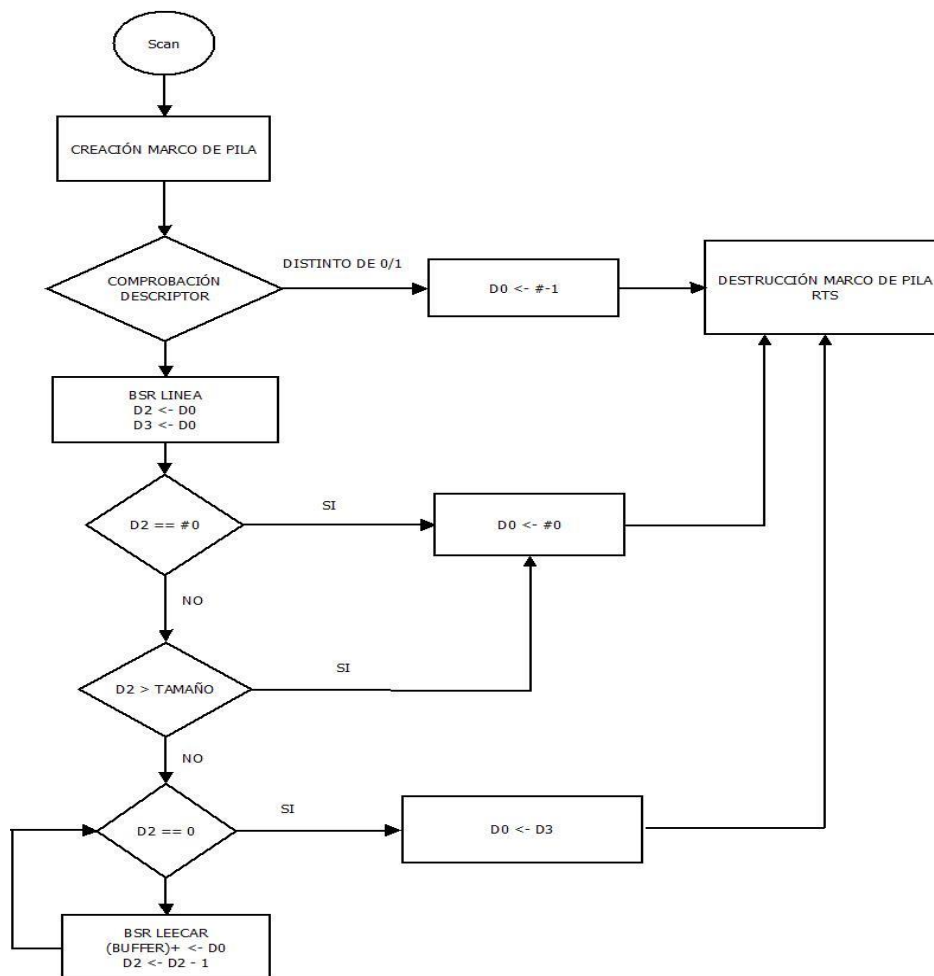


▪ SCAN

```

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;
    }
}

```

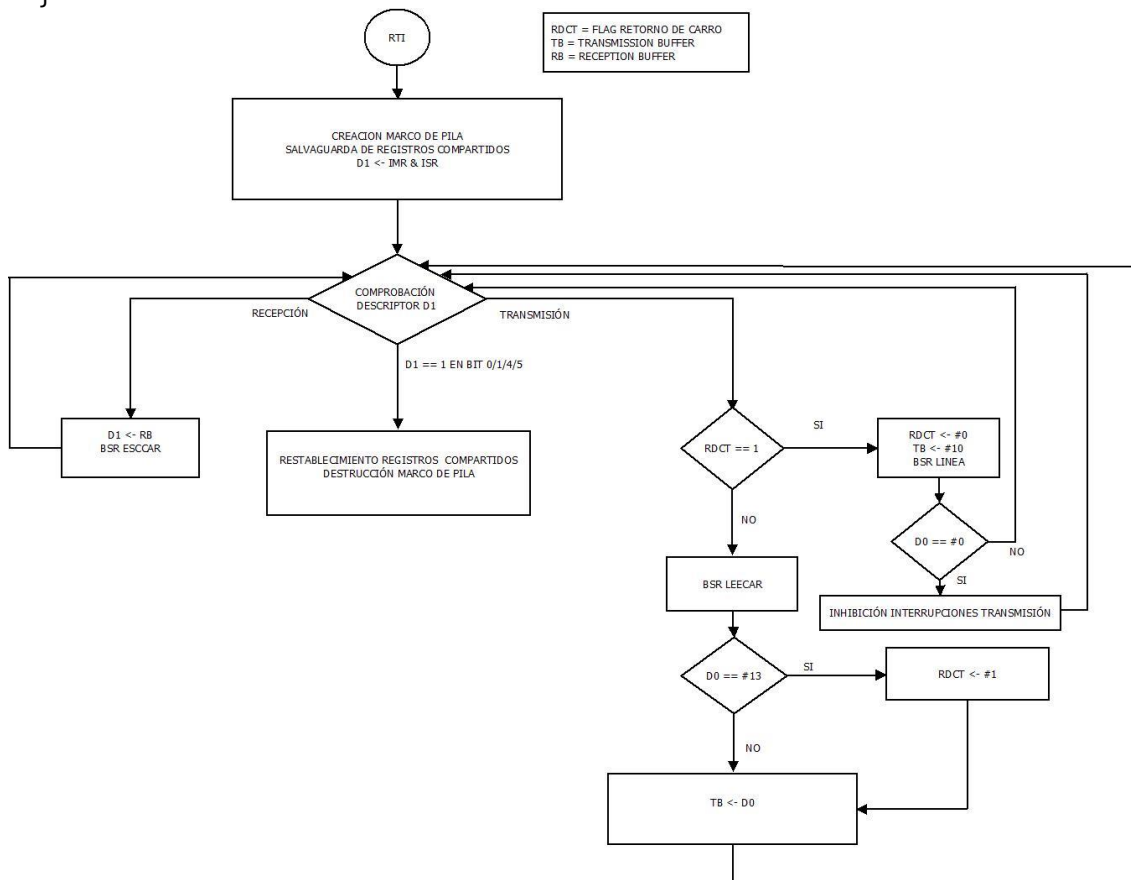


- RTI

```

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());
    }
}

```



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      #%00000000,MR2B     * Eco 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,CRA      * Transmision y recepcion activados.
MOVE.B      #%00000101,CRB      * Transmision y recepcion activados.

MOVE.B      #$00000040,IVR      * Inicializa el vector a hex 40
MOVE.B      #%00100010,IMR      * Inicializa interrupciones
escritura
MOVE.B      #%00100010,IMRC     * Inicializa interrupciones escritura

*Apuntamos los punteros de escritura/lectura a la direccion de su buffer
MOVE.L      #BPA,A0
MOVE.L      A0,PPAE
MOVE.L      A0,PPAL

MOVE.L      #BPB,A0
MOVE.L      A0,PPBE
MOVE.L      A0,PPBL

MOVE.L      #BSA,A0
MOVE.L      A0,PSAE
MOVE.L      A0,PSAL

MOVE.L      #BSB,A0
MOVE.L      A0,PSBE
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)

LEECAR:

AND.L	#3,D0	*guardo los 2 bits mas significativos
CMP.L	#0,D0	*comparo d0 con 00
BEQ	LESA	*LEECAR SCAN A
CMP.L	#1,D0	*comparo d0 con 01
BEQ	LESB	*LEECAR SCAN B
CMP.L	#2,D0	*comparo d0 con 10
BEQ	LEPA	*LEECAR PRINT A
CMP.L	#3,D0	*comparo d0 con 11
BEQ	LEPB	*LEECAR PRINT B

LESA: *LEECAR SCAN A

MOVE.L	#0,D0	
MOVE.L	PSAE,A2	*guardo puntero de escritura de A(scan)
MOVE.L	PSAL,A3	*guardo puntero de lectura de A(scan)
CMP.L	A2,A3	*comparo los punteros-> Miro que los
*puntero de escritura u lectura		
*no esten a la misma altura		
BEQ	FINLEC	*si son iguales devuelve en
*D0=0xFFFFFFFF-> si estan se devuelve		
*-1(buffer vacio)		
MOVE.B	(A3)+,D0	*devuelve en D0 el dato-> sino se
*devuelve el dato y se aumenta el		
*puntero		
MOVE.L	#2001,A4	*guardo 2001 en A4
ADD.L	#BSA,A4	*guardo la direccion final del buffer
CMP.L	A4,A3	*comparo A4,A3-> si no se ha llegado al
*final del buffer se guarda el puntero		
BEQ	FLESA	
MOVE.L	A3,PSAL	*guardo en la direccion el avance del
*puntero		
BRA FINLE		

FLESA: *FIN LEECAR SCAN A

MOVE.L	#BSA,A3	*muevo el puntero a la direccion inicial
*-> Si estamos al final del buffer se		
*pone el puntero al inicio y se guarda		
MOVE.L	A3,PSAL	*guardo en la direccion el avance del
*puntero		
BRA FINLE		

LESB: *LEECAR SCAN B

MOVE.L	#0,D0	
MOVE.L	PSBE,A2	*guardo puntero de escritura de B(scan)
MOVE.L	PSBL,A3	*guardo puntero de lectura de B(scan)
CMP.L	A2,A3	*comparo los punteros
BEQ	FINLEC	*si son iguales devuelve en
*D0=0xFFFFFFFF		
MOVE.B	(A3)+,D0	*devuelve en D0 el dato
MOVE.L	#2001,A4	*guardo 2001 en A4

ADD.A.L	#BSB,A4	*guardo la direccion final del buffer
CMP.L	A4,A3	*comparo A4,A3
BEQ	FLESB	
MOVE.L	A3,PSBL	*guardo en la direccion el avance del
*puntero		
BRA FINLE		
FLESB:	*FIN LEECAR SCAN B	
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:	*LEECAR PRINT A	
MOVE.L	#0,D0	
MOVE.L	PPAE,A2	*guardo puntero de escritura de
*A(print)		
MOVE.L	PPAL,A3	*guardo puntero de lectura de A(print)
CMP.L	A2,A3	*comparo los punteros
BEQ	FINLEC	*si son iguales devuelve en
*D0=0xFFFFFFFF		
MOVE.B	(A3)+,D0	*devuelve en D0 el dato
MOVE.L	#2001,A4	*guardo 2001 en A4
ADD.A.L	#BPA,A4	*guardo la direccion final del buffer
CMP.L	A4,A3	*comparo A4,A3 BEQ
FLEPA		
MOVE.L	A3,PPAL	*guardo en la direccion el avance del
*puntero		
BRA FINLE		
FLEPA:	*FIN LEECAR PRINT A	
MOVE.L	#BPA,A3	*muevo el puntero a la direccion inicial
MOVE.L	A3,PPAL	*guardo en la direccion el avance del
*puntero		
BRA FINLE		
LEPB:	*LEECAR PRINT B	
MOVE.L	#0,D0	
MOVE.L	PPBE,A2	*guardo puntero de escritura de
*B(print)		
MOVE.L	PPBL,A3	*guardo puntero de lectura de B(print)
CMP.L	A2,A3	*comparo los punteros
BEQ	FINLEC	*si son iguales devuelve en
*D0=0xFFFFFFFF		
MOVE.B	(A3)+,D0	*devuelve en D0 el dato
MOVE.L	#2001,A4	*guardo 2001 en A4
ADD.A.L	#BPB,A4	*guardo la direccion final del buffer
CMP.L	A4,A3	*comparo A4,A3
BEQ	FLEPB	
MOVE.L	A3,PPBL	*guardo en la direccion el avance del
*puntero		
BRA FINLE		
FLEPB:	*FIN LEECAR PRINT B	
MOVE.L	#BPB,A3	*muevo el puntero a la direccion inicial

```

        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        #$FFFFFF,D0    *devuelve 0xFFFFFFFF

```

```

FINLE:   *FIN LEECAR

```

```

        RTS

```

▪ ESCCAR(BUFFER,CARÁCTER)

```

ESCCAR:
        AND.W        #3,D0            *me quedo con los dos bits
*psignificativos
        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			
	CMP.L	A0,A1	*A0=A1??
	BEQ	EPAFB	*FINAL DE BUFFER
	CMP.L	A0,A2	
	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 inicial			
	CMP.L	A0,A2	
	BNE	FEPA	
	MOVE.L	#-1,D0	
	BRA FINE		
FEPA:	*FIN ESCCAR PRINT A		
	MOVE.L	A0,PPAE	
	BRA FINE		
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	
	ADDA.L	#2001,A1	*CARGO DIRECCION FINAL DEL
*BUFFER			
	CMP.L	A0,A1	*A0=A1??
	BEQ	ESBFB	*FINAL DE BUFFER
	CMP.L	A0,A2	
	BNE	FESB	
	MOVE.L	#-1,D0	
	BRA FINE		
ESBFB:	*ESCCAR SCAN B FIN DE BUFFER		
	MOVE.L	#BSB,A0	*muevo el puntero a la
*direccion inicial			
	CMP.L	A0,A2	
	BNE	FESB	
	MOVE.L	#-1,D0	
	BRA FINE		
FESB:	*FIN ESCCAR SCAN B		
	MOVE.L	A0,PSBE	
	BRA FINE		
EPB:	*ESCCAR PRINT B		
	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	#BPB,A1	
	ADDA.L	#2001,A1	*CARGO DIRECCION FINAL DEL
*BUFFER			
	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
        BNE           FEPB
        MOVE.L        #-1,D0
        BRA FINE
FEPB:    *FIN ESCCAR PRINT B
        MOVE.L        A0,PPBE
        BRA FINE

FINE:

        RTS

```

▪ LINEA(BUFFER)

LINEA:

```

        AND.W         #3,D0            *Dos bits significativos
        CMP.L         #0,D0
        BEQ           LSA              *LINEA SCAN A
        CMP.W         #2,D0
        BEQ           LPA              *LINEA PRINT A
        CMP.W         #1,D0
        BEQ           LSB              *LINEA SCAN B
        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
        MOVE.L        PSAL,A0          *A0 PUNTERO CARGO PUNTERO DE LECTURA
        MOVE.L        PSAE,A1          *CARGO PUNTERO DE ESCRITURA
BLSA:   *BUCLE LSA
        CMP.L         A1,A0            *MIRO A VER SI ESTA FLV
        BEQ           FLV
        CMP.L         A4,A0
        BEQ           FLSA
        MOVE.B        (A0),D5
        CMP.L         #13,D5           *USAMOS PUNTERO SCAN DE A LECTURA
        BEQ           FINLA
        ADD.L         #1,D3
        ADDA.L        #1,A0            *MUEVO EL PUNTERO
        BRA           BLSA
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
        MOVE.L        PPAE,A1          *CARGO PUNTERO DE ESCRITURA
BLPA:   * BUCLE LINEA PRINT A

```

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

```

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:
        LINK        A6,#0      *CREAMOS MARCO DE PILA

        *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      14(A6),D3   *D1=TAMA?
        CMP.L      #0,D3
        BEQ         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
        BEQ         PRINTB    *escribe en puerto B
        MOVE.L      #-1,D0     *D0=-1 SI NO ES NI 0 NI 1
        BRA         DMPILAPM

```

PRINTA: *PRINT PUERTO A

```

        MOVE.L      #0,D2      *D2=CONTADOR
BUCPA:
        MOVE.L      #2,D0      *PREPARO D0 PARA ESCCAR
MOVE.L    #0,D1      *LIMPIO 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:
        BSR         ESCCAR      *LLAMO A ESCCAR, escribo el caracter
        CMP.L      #-1,D0      *COMPRUEBO VALOR DEVUELTO POR ESCCAR
        BEQ         DMPILAP
        ADD.L      #1,D2      *CONTADOR++
        CMP.L      D2,D3      *MIRO A VER SI HEMOS LLEGADO HASTA
TAMAÑO
        BEQ         DMPILAP

```


	BRA	BUCPA	*SIGUIENTE CARACTER
ACTTA:	BSR	ESCCAR	*LLAMO A ESCCAR
	CMP.L	#-1,D0	*COMPRUEBO VALOR DEVUELTO POR ESCCAR
	BEQ	DMPILAP	
	ADD.L	#1,D2	*CONTADOR++
DMPILAPA:	MOVE.W	#\$2700,SR	*ACTIVO TRANSMISION
	BSET	#0,IMRC	
	MOVE.B	IMRC,IMR	
	MOVE.W	#\$2000,SR	
	BRA	DMPILAP	
PRINTB: *PRINT PUERTO B			
	MOVE.L	#0,D2	*D2=CONTADOR
BUCPB:	MOVE.L	#3,D0	*PREPARO D0 PARA ESCCAR
MOVE.L	#0,D1		*LIMPIO D1
	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	
	ADD.L	#1,D2	*CONTADOR++
	CMP.L	D2,D3	*MIRO A VER SI HEMOS LLEGADO HASTA
*TAMAÑO	BEQ	DMPILAP	
	BRA	BUCPB	
ACTTB:	BSR	ESCCAR	*LLAMO A ESCCAR
	CMP.L	#-1,D0	*COMPRUEBO VALOR DEVUELTO POR ESCCAR
	BEQ	DMPILAP	
	ADD.L	#1,D2	*CONTADOR++
DMPILAPB:	MOVE.W	#\$2700,SR	*ACTIVO TRANSMISION
	BSET	#4,IMRC	
	MOVE.B	IMRC,IMR	
	MOVE.W	#\$2000,SR	
	BRA	DMPILAP	
DMPILAT:	MOVE.L	#0,D2	
DMPILAP:	MOVE.L	D2,D0	*METO EL NUMERO DE CARACTERES ESCRITOS
DMPILAPM:	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	*D1=tamaño
MOVE.W	14(A6),D1	*D0=descriptor
MOVE.W	12(A6),D0	*A0=buffer CARGO EL Buffer
MOVE.L	8(A6),A0	*miro a ver en que puerto va a leer
CMP.W	#0,D0	*escribe en puerto A
BEQ	SCANA	
CMP.W	#1,D0	
BEQ	SCANB	*escribe en puerto B
MOVE.L	#-1,D0	*D0=-1 SI NO ES NI 0 NI 1
BRA	DMPILAS	
SCANA:	*SCAN PUERTO A	
MOVE.B	#0,D0	*D0=0 BUSCAMOS CUANTOS CARACTERES
*HAY EN EL BUFFER DE SCAN A		
BSR	LINEA	
MOVE.L	8(A6),A0	*A0=buffer CARGO EL Buffer
MOVE.L	D0,D2	*D2=LINEA
CMP.W	#0,D2	*LINEA=0?
BEQ	FINCEROA	
MOVE.L	D2,D3	*D3 REGISTRO TAMAÑO ESCRITO
CMP.W	D1,D2	*COMPARO TAMA? Y LINEA
BGT	FINCEROA	
BUCSA:		
CMP.W	#0,D2	*LINEA=0?
BEQ	FINSCANA	
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	
FINCEROA:	MOVE.L #0,D0	*DEVUELVE 0 EN D0
	BRA DMPILAS	
FINSCANA:		
MOVE.L	D3,D0	*D0=N
BRA	DMPILAS	
SCANB:	*SCAN PUERTO B	
MOVE.L	#0,D0	*D0=0
MOVE.L	#1,D0	*LLAMO A LINEA PARA SABER CUAL ES
BSR	LINEA	
*EL TAMAÑO DE LINEA		
MOVE.L	8(A6),A0	*A0=buffer CARGO EL Buffer
MOVE.L	D0,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	
BUCSB:		
CMP.W	#0,D2	*LINEA=0?
BEQ	FINSCANB	
MOVE.L	#1,D0	*PARAMETRO PARA LEECAR
BSR	LEECAR	
MOVE.B	D0,(A0)+	*COPIO EL CARACTER EN BUFFER
SUB.L	#1,D2	*N--
BRA	BUCSB	

```

FINCEROB: MOVE.L    #0,D0                *DEVUELVE 0 EN D0
          BRA DMPILAS
FINSKANB: MOVE.L    D3,D0                *D0=N
          BRA DMPILAS
DMPILAS:

```

```

          UNLK      A6                    *DESTRUCCION MARCO DE PILA
          RTS

```

▪ RTI()

RTI:

```

          LINK      A6,#-36                *Guardamos todos los registros para
*asegurar que no hay problemas de
*conurrencia
          MOVE.L    D0,-4(A6)
          MOVE.L    D1,-8(A6)
          MOVE.L    D3,-12(A6)
          MOVE.L    D5,-16(A6)
          MOVE.L    A0,-20(A6)
          MOVE.L    A1,-24(A6)
          MOVE.L    A2,-28(A6)
          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
          AND.B     ISR,D1                *FUNCION AND entre iSR y imrc
BTST     #0,D1                            *MIRO EL BIT 0 DE D1
          BNE      TA
          BTST     #1,D1                *MIRO EL BIT 1 DE D1
          BNE      RA
          BTST     #4,D1                *MIRO EL BIT 4 DE D1
          BNE      TB
          BTST     #5,D1                *MIRO EL BIT 5 DE D1
          BNE      RB
          BRA      FINREAL

```

TA: *TRANSMISION DE A

```

          CMP.L     #1,RDCTA
          BEQ      FINTA
          MOVE.L    #2,D0                *METO EN D0 3
          BSR      LEECAR
          CMP.L     #13,D0                *RETORNO DE CARRO?
          BNE      VUELTATA
RETCATA:
          MOVE.L    #1,RDCTA
VUELTATA:
          MOVE.B    D0,TBA
          BRA      FINRTI
FINTA:
          MOVE.L    #0,RDCTA

```

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 D0 3
BSR	LEECAR	
CMP.L	#13,D0	*RETORNO DE CARRO?
BNE	VUELTATB	
RETCATB:		
MOVE.L	#1,RDCTB	
VUELTATB:		
MOVE.B	D0,TBB	*NO RETORNO DE CARRO, SI LINEA,
*METO CARACTER		
BRA	FINRTI	
FINTB:		
MOVE.L	#0,RDCTB	
MOVE.B	#10,TBB	*NO MAS LINEAS, SI RET DE CARRO,
*METO SALTO DE LINEA		
MOVE.L	#3,D0	
BSR	LINEA	
CMP.L	#0,D0	*LINEA =0?
BNE	FINRTI	
BCLR	#4,IMRC	*INHIBO INTERRUPCIONES EN TB
MOVE.B	IMRC,IMR	
BRA	FINRTI	
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		
MOVE.L	#0,D1	
MOVE.B	RBB,D1	*CARACTER PARA ESCCAR
MOVE.L	#1,D0	*BUFFER PARA ESCCAR(RBB)
BSR	ESCCAR	
BRA	FINRTI	
FINRTI:		
BRA AMBOS		*COMPRUEBA SI HAY MÁS
*INTERRUPCIONES		

FINREAL:

```
MOVE.L    -36(A6),A4
MOVE.L    -32(A6),A3
MOVE.L    -28(A6),A2
MOVE.L    -24(A6),A1
MOVE.L    -20(A6),A0
MOVE.L    -16(A6),D5
MOVE.L    -12(A6),D3
MOVE.L    -8(A6),D1
MOVE.L    -4(A6),D0
UNLK      A6
RTE
```

*DESTRUCCION MARCO DE PILA

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

▪ Prueba ESCCAR

INICIO:

```
ABUCLEE:
MOVE.L   #1500,D7
```

```

        ADD.L    #1,D6
        MOVE.L   D6,D1
BUCLEE:
        CMP.L    #0,D7
        BEQ      ABUCLEL
        MOVE.L   #2,D0
        BSR      ESCCAR
        SUB.L    #1,D7
        BRA      BUCLEE
ABUCLEL:
        MOVE.L   #1500,D7
BUCLEL:
        CMP.L    #0,D7
        BEQ      ABUCLEE2
        MOVE.L   #2,D0
        BSR      LEECAR
        SUB.L    #1,D7
        BRA      BUCLEL
ABUCLEE2:
        MOVE.L   #999,D7
        BREAK
        ADD.L    #1,D6
        MOVE.L   D6,D1
BUCLEE2:
        CMP.L    #0,D7
        BEQ      ABUCLELI
        MOVE.L   #2,D0
        BSR      ESCCAR
        SUB.L    #1,D7
        BRA      BUCLEE2
ABUCLELI:
        MOVE.L   #13,D1
        MOVE.L   #2,D0
        BSR      ESCCAR
        MOVE.L   #2,D0
        BREAK
        BSR      LINEA

        BREAK

```

```

FIN:    BREAK
BUS_ERROR: BREAK
        NOP
ADDRESS_ER: BREAK
        NOP
ILLEGAL_IN: BREAK
        NOP
PRIV_VIOLT: BREAK

```

```

* Bus error handler
* Address error handler
* Illegal instruction handler
* Privilege violation handler

```

▪ Prueba LINEA

INICIO:

```

        BSR      INIT

        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:       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
TAMPB:        EQU          30           *Tamaño de bloque para PRINT

```

INICIO:

```

*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       #PRIV_VIOLT,32    * Privilege violation handler

```

```

BSR          INIT
MOVE.W       #$2000,SR          *Permite interrupciones

```

```

MOVE.L       #BUFP,A0
MOVE.B       #1,(A0)+
MOVE.B       #2,(A0)+
MOVE.B       #3,(A0)+
MOVE.B       #4,(A0)+
MOVE.B       #5,(A0)+
MOVE.B       #13,(A0)+
MOVE.L       #BUFP,A0
MOVE.W       #0,D0
MOVE.W       #1,D3
MOVE.W       D3,-(A7)
MOVE.W       D0,-(A7)
MOVE.L       A0,-(A7)
BSR          PRINT

```

BREAK

- Prueba 1900 caracteres

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	#\$2000,SR	*Permite interrupciones
MOVE.L	#BUFP,A0	
MOVE.L	#190,D2	*contador
MOVE.L	#0,D4	

BUC:

CMP.L	D4,D2
BEQ	FIN
MOVE.B	#1,(A0)+
MOVE.B	#2,(A0)+
MOVE.B	#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,A0
MOVE.W	#1,D0
MOVE.W	#1901,D3
MOVE.W	D3,-(A7)
MOVE.W	D0,-(A7)
MOVE.L	A0,-(A7)
BSR	PRINT
	BREAK

▪ Prueba SCAN

BUFP:	DS.B	2100	*Buffer para lectura y escritura de caracteres
CONTRP:	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
NLNP:	EQU	1	*Numero de lineas a leer
TAMLP:	EQU	30	*Tamaño de linea para SCAN
TAMPB:	EQU	30	*Tamaño de bloque para PRINT

INICIO:

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

▪ Prueba RTI

BUFFER:	DS.B	2100	*Buffer para lectura y escritura *de caracteres
CONTR:	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 línea para SCAN
TAMB:	EQU	5	*Tamaño de bloque para PRINT
INICIO:			
	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	#PRIV_VIOLT,32	* Privilege violation handler
	BSR	INIT	
	MOVE.W	#\$2000,SR	*Permite interrupciones
BUCPR:	MOVE.W	#0,CONTC	*Inicializa contador de caracteres
	MOVE.W	#NLIN,CONTL	*Inicializa contador de líneas
	MOVE.L	#BUFFER,DIRLEC	*Direccion de lectura (comienzo
del buffer)			
OTRAL:	MOVE.W	#TAML,-(A7)	*Tamaño maximo de la línea
	MOVE.W	#DESA,-(A7)	*Puerto A
	MOVE.L	DIRLEC,-(A7)	*Direccion de lectura
ESPL:	BSR	SCAN	
	CMP.L	#0,D0	
	BEQ	ESPL	*Si no se ha leído una línea de
*intenta de nuevo	ADDA.L	#8,A7	*Restablece la pila
	ADD.L	D0,DIRLEC	*Calcula la nueva dirección de
*lectura			
	ADD.W	D0,CONTC	*Actualiza el número de caracteres
*leídos			
	SUB.W	#1,CONTL	*Actualiza el número de líneas
*leídas. Si no			
	BNE	OTRAL	*se han leído todas las líneas se
*vuelve a leer			
	MOVE.L	#BUFFER,DIRLEC	*Direccion de lectura (comienzo
*del buffer)			
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 dirección del
*buffer			
	SUB.W	D0,CONTC	*Actualiza el contador de caracteres
	BEQ	SALIR	*Si no quedan caracteres se acaba
	SUB.W	D0,TAME	*Actualiza el tamaño de escritura
	BNE	ESPE	*Si no se ha escrito todo el bloque
*se insiste			
	CMP.W	#TAMB,CONTC	*Si el nº de caracteres que quedan
*es menor que el			
			*tamaño establecido se transmite
*ese número			
	BHI	OTRAE	*Siguiendo bloque
	MOVE.W	CONTC,TAME	
	BRA	ESPE	*Siguiendo bloque
SALIR:	BRA	BUCPR	

FIN:
BREAK

BUS_ERROR:
BREAK
NOP

ADDRESS_ER: BREAK
NOP

ILLEGAL_IN:
BREAK
NOP

PRIV_VIOLT:
BREAK
NOP

Observaciones y comentarios