

MANUAL DO PROGRAMADOR
POLY 910 NET

MANUAL DO PROGRAMADOR
POLY 910 NET

CODIGO: 220.390.016-9

ÍNDICE

1 - INTRODUÇÃO	1-1
2 - SISTEMA OPERACIONAL	2-1
2.1 - Carga de Terminal Inteligente sem Unidade de Drive de Disco Flexível	2-1
2.2 - Carga de Terminal Inteligente com Unidade de Drive de Disco Flexível	2-1
2.3 - Arquivos	2-2
2.4 - Endereçamento Lógico dos Drives	2-3
2.5 - Utilitários	2-4
2.5.1 - User	2-4
2.6 - Mensagens de Erro do Sistema	2-5
3 - COBOL 80	3-1
3.1 - Compilação e Linkedição	3-1
3.2 - Arquivos	3-1
3.3 - Biblioteca - "Livronet"	3-1
3.4 - Exemplos	3-4
4 - BASIC 80	4-1
4.1 - Compilação e Linkedição	4-1
4.2 - Arquivos	4-1
4.3 - Biblioteca - "Basnet"	4-1
4.4 - Exemplos	4-4
5 - FORTRAN 80	5-1
5.1 - Compilação e Linkedição	5-1
5.2 - Arquivos	5-1
5.3 - Biblioteca - "Fornet"	5-1
5.4 - Exemplos	5-3
6 - COBOL MB	6-1
6.1 - Diferenças apresentadas pelo COBOL MB	6-1
6.2 - Arquivos	6-1
6.3 - Biblioteca - "Livronet"	6-1

1 - INTRODUÇÃO

O Sistema POLY 910 NET tem como característica básica a interconexão de terminais inteligentes formando uma Rede Local, sem contudo, perder sua independência de processamento.

Assim, o POLY 910 NET é um Sistema que permite a vários usuários o processamento de dados de forma independente em seus respectivos terminais, com acesso comum ou particular a arquivos de trabalho, compondo uma Rede de Processamento Distribuído.

O controle da Rede é realizado através de um gerenciador composto de um microprocessador Z80A e um multiplexador de linha de 8 canais, que gestiona os recursos disponíveis aos diversos terminais, otimizando seu acesso e locação, proporcionando uma facilidade operacional aos usuários.

Com o POLY 910 NET, é possível a programação de um Sistema onde uma integração a nível operativo gerencial se faz necessária, podendo ser configurado de acordo com a necessidade de cada usuário.

Esta sua modularidade permite a conexão de vários tipos de estações terminais, ou seja, microcomputadores com uma CPU, Vídeo/Teclado com uma ou duas unidades de Leitura/Gravação de disquetes, ou terminais inteligentes compostos de uma CPU e Vídeo/Teclado, sem a necessidade de memória de massa secundária.

O objetivo da Rede é partilhar o uso dos periféricos mais caros (Unidade de Disco Rígido, impressora), que serão chamados de periféricos remotos.

Com objetivo de facilitar a referência aos terminais adotaremos as siglas:

- TI - Terminal Inteligente sem unidade de drive de disco flexível.
- TID - Terminal Inteligente com unidade de drive de disco flexível.

É importante lembrar que no desenvolvimento de tal Sistema, procurou-se manter uma compatibilidade a nível de estações terminais aos microcomputadores POLYMAX.

2 - SISTEMA OPERACIONAL

A Rede POLY 910 NET é dividida em dois Sistemas Operacionais:

- . SOP/NET (utilizado na Unidade Gerenciadora), também chamado Sistema Operacional Remoto.
- . SOP/DP (utilizado nos terminais inteligentes com drives-TID), também chamado Sistema Operacional Local.

SOP/NET

O Sistema Operacional Central SOP/NET, é carregado via discos flexíveis da Unidade Gerenciadora, ficando alocado em sua memória RAM durante sua utilização.

Este Sistema Operacional é responsável pelo gerenciamento e controle da Rede, sendo que este processo é feito a nível de alocação de recursos da rede, otimizando e permitindo aos diversos terminais a ela conectados, um acesso de modo mais racional e eficiente possível aos periféricos comuns, ou seja, a impressora de alta velocidade, a(s) Unidade(s) de Disco Rígido, e o(s) drive(s) da Unidade Gerenciadora. Excluindo-se este gerenciamento, o SOP/NET não atua de nenhuma forma sobre o processamento local dos TI ou TID, garantindo-lhes total independência.

As características adicionais deste Sistema Operacional são:

- . Privatização de arquivos via protocolo especial (password)
- . Troca de mensagens (comunicação remota) entre os terminais
- . Impressão pelo modo Spool
- . Sistema de proteção de arquivos e usuários

SOP/DP

O Sistema Operacional Local, SOP/DP, fica residente na memória RAM de cada um dos terminais, sendo carregado de maneira distinta, dependendo do tipo de terminal utilizado (TI ou TID). O SOP/DP é um sistema composto de várias rotinas destinadas ao gerenciamento das operações básicas do Sistema, como organização do espaço em disquete, comunicação do usuário com o terminal e os procedimentos de entrada e saída de dados.

2.1 - CARGA DE TERMINAL INTELIGENTE SEM UNIDADE DE DRIVE DE DISCO FLEXÍVEL

A carga do Sistema Operacional Local (SOP/DP) para este tipo de terminal é feita automaticamente, via gerenciador, tão logo o terminal seja ligado.

Nestas condições o terminal opera sempre no modo ON-LINE, ou seja, ligado à Rede.

2.2 - CARGA DE TERMINAL INTELIGENTE COM UNIDADE DE DRIVE DE DISCO FLEXÍVEL (TID)

A carga do Sistema Operacional Local (SOP/DP), para este tipo de terminal, é feita via disco flexível que deverá ser inserido no drive K.

Uma vez feita esta operação, o terminal estará apto a operar no modo OFF-LINE, ou seja, desconectado da Rede. Neste modo o terminal dispõe de todas as características de um microcomputador isolado. Num segundo momento, em função da necessidade do operador, o terminal pode ser conectado à Rede, através do utilitário C/LIGA4D ou C/LIGA4W, passando a operar no modo ON-LINE, dependendo da configuração apresentada. Para maiores informações, consulte o Manual de Operação do Sistema.

O Sistema de proteção de arquivos, funciona do seguinte modo:

Se estivermos utilizando um OPEN ou um CREATE para um determinado arquivo, quando qualquer outro terminal tentar realizar um OPEN neste mesmo arquivo, o Sistema adverte este segundo terminal de que o arquivo já está aberto e o libera somente para leitura. Se em vez de um OPEN, este terminal tentar realizar um CREATE, ou CLOSE, ou DELETE, ou RENAME, no arquivo já aberto, o Sistema o avisa de que o arquivo já se encontra aberto e cancela a operação.

As mensagens de advertência ou aviso, são seguidas do número do terminal que se encontra alocado.

Para cada usuário existe uma tabela de proteção de arquivos públicos. Essa tabela comporta abertura de 5 arquivos por usuário simultaneamente, com proteção assegurada. No caso de um terminal ter mais do que 5 arquivos abertos sem fechá-los, somente os últimos 5 arquivos abertos ficarão protegidos, ficando os demais desprotegidos.

A proteção a nível de registro permite o acesso ao mesmo arquivo por vários terminais, porém não é feito nenhum controle na proteção de atualização e/ou no mesmo registro. Esse recurso é disponível quando utilizado o COBOL MB (Vide capítulo 6).

A natureza da proteção de arquivos ou registro é geral, ou seja, uma vez escolhida a opção, é válida para todos os terminais.

Para se saber quais arquivos estão abertos (tabela de proteção), basta digitar:

CNTL/D (CR)

e o conteúdo da tabela de proteção será exibido no vídeo.

2.3 - ARQUIVOS

O Sistema POLY 910 NET permite trabalharmos com dois tipos de arquivos:

- . Arquivos Públicos
- . Arquivos Privados ou Particulares

Os arquivos públicos são os que podem ser acessados igualmente por qualquer terminal, enquanto que os arquivos privados são disponíveis somente ao terminal que o criou (através do utilitário USER), estando sempre protegidos. Para utilizar este tipo de arquivo, ver a definição do utilitário USER.

A proteção dos arquivos públicos, também pode ser feita através do utilitário C/CHAVE (senha) ou C/CHAVEW dependendo da configuração utilizada (Disco Rígido Cartridge ou Winchester).

Além deste tipo de proteção, existe ainda uma outra opção para proteção de arquivos remotos (públicos). Esta opção permite realizar proteção a nível de registros ou a nível de arquivos. O tipo de proteção desejado deve ser especificado durante a configuração do Sistema através do disquete SOP/NET - FORMATAÇÃO.

A proteção de arquivos públicos visa tornar um arquivo disponível apenas para leitura para um determinado usuário, mesmo que este arquivo esteja alocado para outro terminal, envolvendo somente arquivos localizados em discos remotos.

2.4 - ENDEREÇAMENTO LÓGICO DOS DRIVES

. Unidade de Disco Rígido Cartridge

No. da Unidade	End. Lógico		End. Lógico	
	Removível	Capacidade	Fixo	Capacidade
Ø	A	5 MB	B	5 MB
1	C	5 MB	D	5 MB
2	E	5 MB	F	5 MB
3	G	5 MB	H	5 MB

. Unidade de Disco Winchester 5MB

No. da Unidade	End. Lógico	Capacidade
Ø	A	5 MB
1	B	5 MB

. Unidade de Disco Winchester 10MB

No. da Unidade	End. Lógico	Capacidade
Ø	A	8 MB
Ø	B	2 MB
1	C	8 MB
1	D	2 MB
2	E	8 MB
2	F	2 MB
3	G	8 MB
3	H	2 MB

. Unidade Gerenciador

No. do Drive	End. Lógico	Capacidade
Ø	I	450 KB
1	J	450 KB

. Terminal com Drive

No. do Drive	End. Lógico	Capacidade	
		dens. simples	dens. dupla
Ø	K	250 KB	450 KB
1	L	250 KB	450 KB

Observação: O endereçamento lógico dos drives abordado neste item é para as versões 4D e 4W do SOP/NET. Quando utilizada a versão 2D o endereçamento dos drives ficará:

A B C D	-	Disco Rígido
E F	-	Gerenciador
G H	-	Terminal com Drive.

2.5 - UTILITÁRIOS

Em complemento aos utilitários apresentados no Manual de Operação do Sistema, temos ainda os seguintes utilitários:

2.5.1 - USER

Aloca uma nova área lógica no mesmo diretório do disco. Cada usuário possui duas opções para este comando no seu terminal:

- USER Ø
- USER N, onde N é o número do seu terminal acrescido de uma unidade.

A numeração lógica dos terminais segue a ordem de Ø a 8. O terminal Ø, por exemplo, pode ter setado seu USER como Ø ou 1. Qualquer outro número de USER resultará na mensagem: "USER INVALIDO".

Sendo assim, cada terminal pode ter arquivos gravados em duas áreas lógicas diferentes em um mesmo disco, sendo acessados somente quando se estiver na mesma área. Na carga do Sistema é assumido USER Ø (default).

Todo arquivo criado no modo USER Ø é chamado ARQUIVO PÚBLICO. Os arquivos criados em outro USER são chamados ARQUIVOS PRIVADOS ou PARTICULARES.

Os arquivos criados no USER de número igual ao terminal + 1, portanto no modo USER particular, serão particulares a este terminal, não sendo possível o acesso a esses arquivos por outros terminais para operações de qualquer natureza.

Isto implica dizer que estes arquivos estão sempre protegidos. O uso destes arquivos se faz do mesmo modo que o dos arquivos PÚBLICOS, e só aparecem no diretório do terminal onde foram criados se estiver no seu modo USER PARTICULAR.

A mudança do tipo de ARQUIVO PÚBLICO para ARQUIVO PRIVADO e vice-versa, é possível através do utilitário PIP com o parâmetro GN.

Portanto, para transferirmos um arquivo de uma área Ø para uma área 1, por exemplo, é necessário que o comando PIP esteja residente na área 1.

EXEMPLO:

Suponha que o terminal 5 está em modo USER PARTICULAR, ou seja, USER 6 (no. terminal + 1), e desejamos transferir todos os arquivos do USER Ø para o USER 6. Para isso, é necessário que o utilitário PIP esteja no diretório do USER 6.

A forma para se salvar o PIP no USER 6, desde que exista o utilitário PIP no USER Ø, é:

```
A > USER Ø (CR)
A > PIP (CR)
*   (CR)
A > USER 6 (CR)
A > SAVE 32 PIP.COM (CR)
```

Uma vez que o PIP foi salvo no USER particular, devemos digitar:

```
PIP A:=A:*. * [GØ] (CR)
```

e teremos a transferência de todos os arquivos do USER Ø para o USER 6.

2.6 - MENSAGENS DE ERRO DO SISTEMA

Além das mensagens de erro usuais emitidas pelo CP/M, o Sistema Operacional da Rede (SOP/NET) envia as seguintes mensagens:

- a) ERRO DE CARGA, REPETIR?
Esta mensagem é emitida quando ocorre erro de carga do Sistema (CCP + BDOS) no "WARM BOOT". Ex: O disquete não está no drive corrente.
- b) ERRO DE SEL. DISCO
Ocorre quando um drive não pertencente ao Sistema for selecionado.
- c) ERRO DE L/E NO DISCO REMOTO
Mensagem enviada quando ocorre erro de acesso a um disco remoto, podendo ser por exemplo um disco não formatado, disco ruim, etc..
- d) ERRO DE COM.:L/E
Erro de leitura ou gravação no disco, ocasionado por alguma falha no meio do processo. Geralmente ocorre por falha no meio de comunicação.
- e) ERRO DE DEN/FMT NO DISQUETE
Erro devido a diferença de densidade do disquete. No caso de terminais inteligentes com drive, o segundo drive deve ser setado com a mesma densidade que o primeiro.
- f) ERRO DE POS.TRILHA = Ø NO DISQUETE
Ocorre quando não se consegue realizar um "home" (posicionamento sobre a trilha Ø).
- g) DISQUETE RUIM. RELOCÁ-LO
Mensagem enviada quando um drive é selecionado e o disco não está colocado corretamente, ou não se encontra no drive.
- h) ERRO DE L/E NO DISQUETE, REPITA OPER.
Mensagem emitida quando é detectado um erro de acesso ao disquete local.
- i) ESCRITA EM ARQ. PROTEGIDO
Ocorre quando houver uma tentativa de escrita sobre um arquivo aberto em outro usuário, estando setada a proteção a nível de arquivos no gerenciador.
- j) ABERTO → X
Ocorre quando acessar um arquivo que está aberto para o terminal X, ou quando se acessar o disquete remoto (do gerenciador) e outro terminal o está acessando.

k) E/C:FNC X

Erro de comunicação gerado durante o processo que trata da função X, onde X pode ser open, close, write sequential, random, read sequential/random, make, delete/rename, etc...

l) ARQ. PROTEGIDO → PASSWORD

Erro provocado pela tentativa de leitura/gravação em um arquivo que está protegido por uma senha.

m) USER INVÁLIDO

Erro devido a tentativa de utilizar um número no comando USER diferente do permitido.

3 - COBOL 80

3.1 - COMPILAÇÃO E LINKEDIÇÃO

Em termos de compilação e linkedição simultânea nos discos remotos, (com dois usuários cada um em terminal, compilando programas fontes diferentes), só será possível se tivermos dois compiladores ou dois linkeditores, em discos remotos diferentes ou operando em USER's diferentes.

O compilador COBOL 80 gera um arquivo intermediário com o nome STEXT.INT, no disco remoto no drive corrente, causando durante uma compilação simultânea uma mistura de programas.

CONCLUSÃO:

No modo USER Ø, a compilação e a linkedição de programas não poderão ser feitas por mais de um usuário ao mesmo tempo, utilizando o mesmo compilador e o mesmo link.

3.2 - ARQUIVOS

Existem dois tipos de arquivos tratados pelo SOP/NET: arquivos públicos ou privados.

A proteção de arquivos públicos (tabela de proteção), objetiva tornar a gravação liberada para o primeiro terminal a abrir o arquivo (OUTPUT), enquanto que todos os outros usuários só poderão ler o arquivo (INPUT).

Algumas precauções devem ser tomadas pelo usuário na utilização de arquivos nos discos remotos através do COBOL 80:

1. Pré-formatar os arquivos. Vantagem: Reservar uma área para o arquivo, impedindo que sejam gravados outros arquivos nesta área pré-definida.
2. Os arquivos antes de qualquer operação de leitura/gravação deverão ser abertos para leitura (OPEN INPUT) e em seguida o "FILE STATUS" do arquivo deverá ser testado:

SE = '91' leitura habilitada (READ ARQUIVO)

SE = '3Ø' Arquivo não existe ou está protegido, ou seja, arquivo privado.

SE = 'ØØ' Arquivo existe e pode ser acessado para regravação (OPEN I-O).

3.3 - BIBLIOTECA - 'LIVRONET'

Este item descreve as rotinas especiais que foram desenvolvidas para o POLY 910 NET sendo usadas no COBOL 80.

Essas rotinas estão embutidas em um único módulo, que deverá estar residente em disco na montagem. Dependendo da configuração utilizada, este deverá ser "LIVRONET.REL" para Disco Rígido Cartridge e "COBNETW.REL" para disco Winchester.

As chamadas dessas rotinas serão descritas a seguir, devendo para sua montagem:

- . Estar com o programa principal em código relocável (.REL) no disco
- . Conter o programa L80.COM no disco.
- . Conter o módulo LIVRONET.REL ou COBNETW.REL no disco utilizando o seguinte formato:

L8Ø D:NOME1/ N, D:NOME2, D:LIVRONET/S/E
ou

L8Ø D:NOME1/ N, D:NOME2, D:COBNETW/S/E ; onde:

D:NOME1 - é o drive e o nome do programa final em código executável (.COM)
D:NOME2 - é o drive e o nome do programa principal em código relocável.
D:LIVRONET - drive e o módulo LIVRONET.REL
D:COBNETW - drive e o módulo COBNETW.REL

/E - montar e terminar

/G - montar e executar

Esta biblioteca possui as seguintes rotinas:

A - SPOOL

Esta rotina monta um arquivo SPOOL no disco remoto, ao invés de enviar diretamente para a impressora. O gerenciador se encarrega de enviar para a impressora remota (Operação de DESPOOL) o arquivo criado no disco.

Considerações para uso da rotina:

A.1. Esta rotina deve ser chamada no início e final de um programa. O arquivo Impressora deverá ser aberto antes do CALL "SPOOL" e fechado antes do CALL "SPOOL" para fechamento.

A.2. CHAVE - Único parâmetro passado para rotina Spool através do COBOL.
Definir como: 77 CHAVE PIC 9.

O conteúdo da CHAVE deve ser:

SE CHAVE = Ø significa abertura para SPOOL (OPEN).

SE CHAVE = 1 significa fechamento do SPOOL (Fim de Serviço).

SE CHAVE = 2 significa fechamento do SPOOL (CLOSE) intermediário.

SE CHAVE = 3 significa reinício de serviço.

A.3. Durante a operação de inicialização ou operação do WRITE na impressora, é colocado na chave (STATUS) o código de erro.

Códigos de erro:

STATUS	SIGINIFICADO
Ø	Operação bem sucedida
1	Arquivo Spool já existe
2	Diretório Cheio
3	Remoto não está pronto (gerenciador)
4	Disco Cheio
5	Impressora não está pronta
6	Erro de comunicação
7	Arquivo Spool não existe.

A.4. Chamada pelo COBOL:

```
CALL "SPOOL" USING CHAVE
```

B - SENHA

Esta rotina permite ao usuário que está utilizando o COBOL 80, acessar a tabela de PASSWORD (tabela de proteção a nível do usuário), isto é, inserir, retirar ou permitir acesso aos arquivos.

Considerações para uso da rotina:

B.1. PALAVRA-CHAVE

Deve conter a senha para o arquivo.

Definir: 77 PALAVRA-CHAVE PIC X (8)

B.2. NOME-ARQUIVO

Nome do arquivo que será acessado. Definir:

Ø1 NOME-ARQUIVO

Ø2 DRIVE PIC X

Ø2 NOME PIC X (8)

B.3. CODIGO

Definir como: 77 CODIGO PIC 9

O conteúdo do código pode ser:

CODIGO = 1 SETAR PASSWORD

CODIGO = 2 APAGAR PASSWORD

CODIGO = 3 HABILITAR ACESSO NO ARQUIVO

CODIGO = 4 DESABILITAR ACESSO AOS ARQUIVOS

B.4. Os códigos de erro serão fornecidos como retorno no código:

Código	Significado
Ø	operação bem sucedida
1	parâmetro código inválido
2	erro no nome do arquivo
3	overflow na tabela de password
4	erro de escrita na tabela
5	arquivo sem password
6	password inválida para arquivo
7	erro de leitura na tabela de password
8	arquivo já existe na tabela de password

B.5. Chamada pelo COBOL

```
CALL "SENHA" USING CODIGO, NOME-ARQUIVO, PALAVRA-CHAVE
```

Se CODIGO = 4 utilizar a seguinte sintaxe:

```
CALL "SENHA" USING CODIGO
```

Não é necessário o nome do arquivo e a palavra chave.

3.4 - EXEMPLOS

A seguir temos dois exemplos de programas COBOL:

A - IMPRESS.COB

B - SENHA.COB

que utilizam as rotinas SPOOL e SENHA, respectivamente.

```
FILE:IMPRESS    COB
IDENTIFICATION DIVISION.
PROGRAM-ID.      "IMPRE".
DATE-WRITTEN.    02/02/83.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    DECIMAL-POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT ARQUIVO ASSIGN TO PRINTER.
    SELECT ETIQUETA ASSIGN TO DISK
    ORGANIZATION IS RELATIVE
    ACCESS MODE IS RANDOM
    RELATIVE KEY IS CHAVE
    FILE STATUS IS POS.
```

```

DATA DIVISION.
FILE SECTION.
FD ARQUIVO
    LABEL RECORD IS OMITTED.
Ø1 BUF PIC X(132).
FD ETIQUETA
    LABEL RECORD IS STANDARD
    VALUE OF FILE-ID IS "A:ETIQUETA".
Ø1 AREARAN.
    Ø2 PAG PIC 9(5).
    Ø2 PAG2 PIC 9(5).
WORKING-STORAGE SECTION.
77 POS PIC XX VALUE SPACES.
77 PARAM PIC 9 VALUE Ø.
77 CHAVE PIC 99999 COMP VALUE 1.
77 CONTADOR1 PIC 9(5) VALUE Ø.
77 CONTADOR2 PIC 9(5) VALUE Ø.
Ø1 REG.
    Ø2 CAMPO11 PIC 99999 VALUE ZEROS.
    Ø2 CAMPO12 PIC X(5Ø) VALUE ALL "#".
    Ø2 CAMPO13 PIC X(5ØØ) VALUE ALL "*".
    Ø2 CAMPO14 PIC X(27) VALUE ALL "+".
Ø1 CABEC.
    Ø2 FILLER PIC X(55) VALUE SPACES.
    Ø2 FILLER PIC X(7) VALUE "PAG. ".
    Ø2 PAGINA PIC 9(5) VALUE Ø.
    Ø2 FILLER PIC X(65) VALUE SPACES.
Ø1 NOME-ARQUIVO.
    Ø2 DRIVE PIC X.
    Ø2 NOME PIC X(8).
    Ø2 TIPO PIC X(3).
PROCEDURE DIVISION.
INICIO.
    DISPLAY (1, 1) ERASE.
    DISPLAY (5, 7) "DRIVE...< > ".
    DISPLAY (, ) "NOME...< > ".
    DISPLAY (5, 43) "TIPO...< > ".
    ACCEPT (5, 17) NOME WITH AUTO-SKIP.
    ACCEPT (5, 31) TIPO WITH AUTO-SKIP.
    ACCEPT (5, 52) TIPO WITH AUTO-SKIP.
    OPEN OUTPUT ARQUIVO.
    DISPLAY (1, 1) ERASE.
    OPEN INPUT ETIQUETA.
    IF POS = "ØØ"
        READ ETIQUETA
        GO TO REINICIO.

```

```
LIGA.  
  OPEN OUTPUT ETIQUETA.  
  MOVE Ø TO PAG PAG2.  
  WRITE AREARAN.  
  CLOSE ETIQUETA.  
ACIL.  
  MOVE Ø TO PARAM.  
  CALL "SPOOL" USING PARAM.  
  IF PARAM NOT = Ø GO TO ERRO.  
VOLTA.  
  MOVE SPACES TO BUF.  
  PERFORM CABECALHO THRU FIM-CABEC.  
LOOP1.  
  ADD 1 TO CONTADOR1.  
  ADD 1 TO CONTADOR2.  
  MOVE CONTADOR1 TO CAMPO11.  
  WRITE BUF FROM REG AFTER ADVANCING 1 LINES.  
  IF PAGINA > 3Ø  
  GO TO FINAL.  
  IF CONTADOR2 > 5Ø  
    PERFORM CABECALHO THRU FIM-CABEC  
    MOVE Ø TO CONTADOR2.  
  IF PAGINA > 1  
  OPEN I-O ETIQUETA  
  READ ETIQUETA  
  MOVE PAGINA TO PAG  
  MOVE CONTADOR1 TO PAG2  
  REWRITE AREARAN  
  CLOSE ETIQUETA  
  MOVE 2 TO PARAM  
  CALL "SPOOL" USING PARAM  
  IF PARAM NOT =Ø GO TO ERRO.  
  DISPLAY (12, 3Ø) "PAGINA ..." PAGINA.  
  GO TO LOOP1.  
ERRO.  
  DISPLAY (1, 1) ERASE  
  DISPLAY (12, 2Ø) "ERRO - PARAMETRO = <"PARAM"> "  
  STOP RUN.  
CABECALHO.  
  ADD 1 TO PAGINA.  
  MOVE ALL "*" TO BUF.  
  WRITE BUF AFTER ADVANCING PAGE.  
WRITE BUF FROM CABEC AFTER ADVANCING 1 LINES.  
FIM-CABEC.  
  EXIT.
```

```
REINICIO.
  MOVE PAG TO PAGINA.
  MOVE PAG2 TO CONTADOR1.
  SUBTRACT 1 FROM PAGINA.
  MOVE 3 TO PARAM.
  CALL "SPOOL" USIN PARAM.
  IF  PARAM = 7
    DISPLAY (1, 1) ERASE
    DISPLAY (12, 40) "ARQUIVO SPOOL NAO EXISTE"
    STOP RUN.
  PERFORM CABECALHO THRU FIM-CABEC.
  MOVE 0 TO CONTADOR2.
  CLOSE ETIQUETA.
  GO TO LOOP1.
FINAL.
  CLOSE ARQUIVO.
  MOVE 1 TO PARAM.
  CAL "SPOOL" USING PARAM.
  IF PARAM NOT = 0 GO TO ERRO.
STOP RUN.
```

FILE: SENHA COB

```
IDENTIFICATION DIVISION.
PROGRAM-ID.        "PASSE".
DAE-WRITTEN.      17/02/83.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
  DECIMAL-POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT ETIQUETA ASSIGN TO DISK
    ORGANIZATION IS RELATIVE
    ACCESS MODE IS RANDOM
    RELATIVE KEY IS CHAVE
    FILE STATUS IS POS.
DATA DIVISION.
FILE SECTION.
FD        ETIQUETA
  LABEL RECORD IS STANDARD
  VALUE OS FILE-ID IS VARIABEL.
01        AREARAN.
  02 PAG        PIC 9(5).
  02 PAG2       PIC 9(5).
```

WORKING-STORAGE SECTION.

```

77     POS          PIC XX VALUE SPACES.
77     PARAM        PIC 9 VALUE Ø.
77     INCHAR       PIC X.
77     OPCA0        PIC 9 VALUE Ø.
77     CHAVE        PIC 99999 COMP VALUE 1.
77     CONTADOR1    PIC 9(5) VALUE Ø.
77     PASSE        PIC X(8).
77     CONTADOR2    PIC 9(5) VALUE Ø.

```

```

Ø1     VARIABEL.
      Ø2 DRIVE1 PIC X.
      Ø2 FILLER PIC X VALUE ":".
      Ø2 NOME1 PIC X(8).

```

```

Ø1     NOME-ARQUIVO.
      Ø2 DRIVE  PIC X.
      Ø2 NOME   PIC X(8).

```

SCREEN SECTION.

```

Ø1     SEGURO SECURE.
      Ø2 LINE 12 COLUMN 26 PIC X(8) TO PASSE.

```

PROCEDURE DIVISION.

INICIO.

```

      DISPLAY "DIGITAR NOME DO ARQUIVO A SER CRIADO...".
      ACCEPT DRIVE1.
      ACCEPT NOME1.
      OPEN OUTPUT ETIQUETA.
      MOVE Ø TO PAG PAG2.
      WRITE AREARAN.
      CLOSE ETIQUETA.

```

MENU.

```

      DISPLAY (1, 1) ERASE.
      DISPLAY (5, 5) "SETAR PASSWORD.....1".
      DISPLAY (7, 5) "DELETAR PASSWORD.....2".
      DISPLAY (9, 5) "HABILITAR ACESSO AO ARQUIVO.....3".
      DISPLAY (11, 5) "DESABILITAR ACESSO AO ARQUIVO....4".
      DISPLAY (13, 5) "ACESSO AO ARQUIVO SEM PASSWORD...5".
      DISPLAY (15, 5) "FIM DE SERVIÇO.....6".
      DISPLAY (17, 5) "DIGITAR OPCA0.....< > ".
      ACCEPT (17, 38) OPCA0.
      DISPLAY (1, 1) ERASE.

```

LICA1.

```

      MOVE OPCA0 TO PARAM.
      IF OPCA0 = 4
          GO TO DESABILITA.
      IF OPCA0 = 6 GO TO FIM.
      DISPLAY (4, 7) "NOME DO ARQUIVO".
      DISPLAY (5, 7) "DRIVE...< >".
      DISPLAY ( , ) "NOME...<           >".
      ACCEPT (5, 17) DRIVE WITH AUTO-SKIP.
      ACCEPT (5, 31) NOME WITH AUTO-SKIP.
      MOVE NOME TO NOME1.
      MOVE DRIVE TO DRIVE1.
      IF OPCA0 = 5 GO TO LICA2.
      DISPLAY (12, 2) "DIGITAR PASSWORD...<           >".
      ACCEPT SEGURO.
      CALL "SENHA" USING PARAM, NOME - ARQUIVO, PASSE.

```

```
SALAO.
    DISPLAY PARAM.
    IF PARAM NOT = Ø GO TO ERRO.
    IF OPCA0 = 1 OR OPCA0 = 4 GO TO MENU.
LICA2.
    DISPLAY (1, 1) ERASE.
    OPEN INPUT ETIQUETA.
    DISPLAY "OPEN STATUS" POS.
    ACCEPT (, ) INCHAR.
    IF POS NOT = "ØØ" GO TO ERRO.
    READ ETIQUETA.
    DISPLAY "READ STATUS" POS.
    ACCEPT (, ) INCHAR.
    IF POS NOT = "ØØ" GO TO ERRO.
    CLOSE ETIQUETA.
    GO TO MENU.
DESABILITA.
    CALL "SENHA" USING PARAM.
    GO TO SALAO.
ERRO .
    DISPLAY (1, 1) ERASE.
    DISPLAY (7, 2Ø) "ERRO - OPCA0 = < "OPCA0"> ".
    DISPLAY (12, 2Ø) "ERRO - PARAMETRO = < "PARAM"> ".
FIM.
    STOP RUN.
```


4 - BASIC 80

4.1 - COMPILAÇÃO E LINKEDIÇÃO

O Compilador, o interpretador e o link do Basic 80 não criam arquivo intermediário, portanto sua utilização no Sistema POLY 910 NET não possui restrições.

4.2 - ARQUIVOS

Operações de arquivos no Basic 80 possuem as mesmas características gerais dos arquivos no Sistema POLY 910 NET.

Algumas precauções devem ser tomadas pelo usuário na utilização de arquivos nos discos remotos através do BASIC 80:

1. Pré-formatar os arquivos. Vantagem: Reservar uma área para o arquivo, impedindo que sejam gravados outros arquivos nesta área pré-definida.
2. O Sistema protege os arquivos que estão abertos para gravação, enviando mensagens ao usuário que tentar gravar num arquivo já aberto para gravação.

OBSERVAÇÃO 1:

O usuário deve verificar se arquivo que deseja trabalhar (para gravação) está aberto para outro terminal. Solução: utilizar o comando "CNTL D" que mostrará no console os arquivos que estão abertos naquele momento para cada terminal.

OBSERVAÇÃO 2:

O comando NAME do interpretador que modifica o nome de um arquivo em disco possui a seguinte sintaxe:

NAME "<NOME ARQUIVO ANTIGO>" AS "<NOME ARQUIVO NOVO>"

No Sistema POLY 910 NET este comando terá a seguinte restrição:

O nome dos drives dos arquivos antigo e novo devem ser especificados mesmo que o drive dos arquivos seja o drive corrente.

NAME "<D: NOME ARQUIVO ANTIGO>" AS "<D: NOME ARQUIVO NOVO>"

D = Drive do arquivo.

4.3 - BIBLIOTECA - "BASNET"

Este item descreve as rotinas especiais que foram desenvolvidas para o POLY 910 NET, sendo utilizadas no BASIC 80. Essas rotinas estão embutidas em um único módulo. Dependendo da configuração utilizada, este deve ser "BASNET.REL" para Disco Rígido Cartridge ou "BASNETW.REL" para disco Winchester, podendo ser utilizadas pelo interpretador.

As chamadas dessas rotinas serão descritas a seguir, devendo, para sua montagem, o usuário:

- . Estar com o programa principal em código relocável (.REL) no disco.
- . Conter o programa L80.COM no disco
- . Conter o módulo BASNET.REL ou BASNETW.REL no disco; devendo utilizar o seguinte formato:

L80 D:NOME1/N, D:NOME2, D:BASNET/S/

ou

L80 D:NOME1/N, D:NOME2, D:BASNETW/S/E ; onde:

D:NOME1 - é o drive e o nome do programa final em código executável (.COM)

D:NOME2 - é o drive e o nome do programa principal em código relocável

D:BASNET - drive e módulo BASNET.REL ou BASNETW.REL

/E - montar e terminar

/G - montar e executar

Esta biblioteca possui as seguintes rotinas:

A. SPOOL

Esta rotina gera o arquivo que seria enviado para a impressora remota, no disco remoto (operação Spool).

O gerenciador se encarrega de enviar para a impressora remota (operação despool) o arquivo criado em disco. Essa impressão ocorre quando o índice de uso da Rede cair abaixo de certo nível, (para que a Rede não degrade).

Considerações para uso da rotina:

A.1. Esta rotina deve ser chamada no início e final de um programa.

A.2. CHAVE

Único parâmetro passado para a rotina SPOOL através do BASIC. Definir como: CHAVE% (variável inteira).

O conteúdo da chave pode ser:

SE CHAVE% = 48 significa ABERTURA PARA SPOOL (OPEN)

SE CHAVE% = 49 significa FECHAMENTO DO SPOOL (Fim de Serviço)

SE CHAVE% = 50 significa FECHAMENTO DO SPOOL (CLOSE) intermediário

SE CHAVE% = 51 significa REINICIO DE SERVIÇO

A.3. Durante a operação de inicialização ou operação de WRITE na impressora, é colocado na CHAVE% (Status) o código de erro.

Códigos de erro:

STATUS	SIGNIFICA
48	operação bem sucedida
49	arquivo Spool já existe
50	diretório cheio
51	remoto não está pronto (gerenciador)
52	disco cheio
53	impressora não está pronta
54	erro de comunicação
55	arquivo Spool não existe

A.4. Utilizar a instrução LPRINT normalmente

A.5. Chamada pelo BASIC: CALL SPOOL (CHAVE%)

B. SENHA

Esta rotina permite ao usuário que está utilizando o BASIC 80 (não pelo interpretador MBASIC) no Sistema POLY 910 NET, acessar a tabela de Password (tabela de proteção a nível de usuário), isto é, inserir, retirar ou permitir acesso aos arquivos.

Considerações para uso da rotina: -

B.1. PALAVRACHAVE -

Deve conter a senha do arquivo a ser acessado. Definir:
PALAVRACHAVE\$ (variável String de tamanho 8)

B.2. NOMEARQUIVO -

O conteúdo será o nome do arquivo que irá ser acessado.
Definir: NOMEARQUIVO\$ (variável String contendo 9 caracteres, sendo o primeiro o nome do Drive e os seguintes o nome do Arquivo).

B.3. CODIGO -

Definir como: CODIGO% (variável inteira)

O conteúdo do código pode ser:

```
CODIGO% = 49   SETA PASSWORD
CODIGO% = 50   APAGA PASSWORD
CODIGO% = 51   HABILITA ACESSO NO ARQUIVO
CODIGO% = 52   DESABILITA ACESSO AOS ARQUIVOS
```

B.4. Códigos de erro serão fornecidos como retorno no código:

CÓDIGO	SIGNIFICADO
48	operação bem sucedida
49	parâmetro código inválido
50	erro no nome do arquivo
51	overflow na tabela de password
52	erro de escrita na tabela
53	arquivo sem password
54	password inválido para o arquivo
55	erro de leitura na tabela de password
56	arquivo já existe na tabela de password

B.5. Chamada pelo BASIC

```
CALL SENHA (CODIGO%, NOMEARQUIVO$, PALAVRACHAVE$)
```

Se CODIGO% = 52, utilizar a seguinte sintaxe:

```
CALL SENHA (CODIGO%)
```

Não é necessário o nome do arquivo e a palavra chave.

4.4 - EXEMPLOS:

A seguir, temos dois exemplos de programas BASIC:

A - IMPRBAS.BAS

B - SENHABAS.BAS

que utilizam as rotinas SPOOL e SENHA, respectivamente.

FILE : IMPRBAS BAS

```
10 ON ERROR GOTO 150
20 OPEN "I", #1, "DATA"
30 CHAVE%=51
40 GOSUB 270
50 INPUT #1,NP$
60 X$= STRING$(129, "*"): CAB$=X$+NP$
70 LPRINT CAB$:FOR I=1 TO 50:LPRINT "*":NEXT I
80 LPRINT CHR$(12)
90 CLOSE #1
100 PAG%VAL(NP$)
110 IF PAG%=30 THEN GOTO 240
120 CHAVE%=50
130 GOSUB 270
140 GOTO 180
150 PAG%=0:CLOSE #1
160 CHAVE%=48
170 GOSUB 270
180 OPEN "O", #1, "DATA"
190 PAG%=PAG%+1
200 NP$=SRT$(PAG%)
210 WRITE #1, NP$
220 PRINT "NUMERO DA PAGINA: ", NP$
230 GOTO 60
240 CHAVE%=49
250 GOSUB 270
260 GOTO 300
270 CALL SPOOL (CHAVE%)
280 IF CHAVE%<>48 THEN PRINT "ERRO - PARAMETRO = ", CHAVE%:GOTO 300
290 RETURN
300 END
```

FILE: SENHA BAS

```
10 INPUT "DIGITE NOME-DRIVE: "; D$
20 D$=LEFT$(D$+" ",1)
30 INPUT "DIGITE NOME-ARQUIVO: "; A$
40 A$ = LEFT$(A$ + " ", 8)
50 A$ = D$ + A$
60 INPUT "DIGITE SENHA: "; S$
70 S$ = LEFT$(S$ + " ", 8)
80 PRINT "SETAR PASSWORD.....(49)"
90 PRINT "APAGAR PASSWORD.....(50)"
100 PRINT "HABILITA ACESSO AO ARQUIVO....(51)"
110 PRINT "DESABILITA ACESSO AO ARQUIVO..(52)"
120 PRINT "FIM DE SERVICO.....(53)",
130 INPUT "DIGITE OPCA.....", C%
140 IF C%=53 GOTO 190
150 CALL SENHA(C%, A$, S$)
160 IF C$<>48 GOTO 180
170 GOTO 10
180 PRINT "ERRO: "; C%
190 END
```


5 - FORTRAN 80

5.1 - COMPILAÇÃO E LINKEDIÇÃO

Em termos de compilação e linkedição simultânea nos discos remotos, (com dois usuários um em cada terminal, compilando programas- -fonte), só será possível se tivermos dois compiladores ou dois linkeditores, em discos remotos diferentes ou operando em USER's diferentes.

5.2 - ARQUIVOS

As operações de arquivos no FORTRAN 80, possuem as mesmas características gerais dos arquivos no Sistema POLY 910 NET. Algumas precauções devem ser tomadas pelo usuário na utilização de arquivos nos discos remotos através do FORTRAN 80:

1. Pré-formatar os arquivos. Vantagem: Reservar uma área para o arquivo, impedindo que sejam gravados outros arquivos nesta área pré-definida.
2. O usuário deverá verificar se o arquivo que deseja trabalhar (para gravação) está aberto para outro terminal. Solução: Utilizar o comando "CNTL D", que exibirá no console os arquivos que estão abertos naquele momento para cada terminal.

5.3 - BIBLIOTECA "FORNET"

Este item descreve as rotinas especiais que foram desenvolvidas para o POLY 910 NET sendo utilizadas no FORTRAN 80.

Essas rotinas estão embutidas em um único módulo. Dependendo da configuração utilizada, este deve ser o "FORNET.REL" para Disco Rígido Cartridge ou "FORNETW.REL" para disco Winchester.

As chamadas dessas rotinas serão descritas a seguir. Para sua montagem o usuário deverá:

- . Estar com o programa principal em código relocável (.REL) no disco.
- . Conter o programa L80.COM no disco.
- . Conter o módulo FORNET.REL ou FORNETW.REL no disco, devendo utilizar o seguinte formato:

L80 D:NOME1/N, D:NOME2, D:FORNET/S/E

ou

L80 D:NOME1/N, D:NOME2, D:FORNETW/S/E ; onde:

D:NOME1 - é o drive e o nome do programa final em código executável (.COM)

D:NOME2 - é o drive e o nome do programa principal em código relocável

D:FORNET - drive e módulo FORNET.REL ou FORNETW.REL

/E - montar e terminar

/G - montar e executar

Esta biblioteca possui as seguintes rotinas:

A. SPOOL

Esta rotina quando chamada, cria arquivo spool no disco remoto. O gerenciador se encarrega de enviar para a impressora remota (operação de DESPOOL) o arquivo criado no disco, quando o índice da rede cair abaixo de certo nível, para não degradar a rede.

Considerações para o uso da rotina:

A.1. Esta rotina deve ser chamada no início do programa. Deverá constar no final do programa o comando END que irá finalizar o programa e a operação de spool (fechando-o).

A.2. O único parâmetro passado para rotina SPOOL através do FORTRAN, é o K onde: K é uma variável inteira de uma posição.

O conteúdo da variável K deve ser igual à zero.

A.3. Durante a operação de inicialização ou operação do WRITE (LUN=2) na impressora, é devolvido na variável K o status da operação.

Códigos de erro:

STATUS	SIGNIFICADO
Ø	operação bem sucedida
1	arquivo SPOOL já existe
2	diretório cheio
3	remoto não está pronto (gerenciador)
4	disco cheio
5	impressora não está pronta
6	erro de comunicação

A.4. Chamada pelo FORTRAN:

K=Ø

CALL SPOOL (K)

B. SENHA

Esta rotina permite ao usuário que está utilizando o FORTRAN 80 no Sistema POLY 910 NET, acessar a tabela de PASSWORD (tabela de proteção à nível de usuário), isto é, inserir, retirar ou permitir acesso aos arquivos.

B.1. Chamada pelo FORTRAN:

CALL SENHA (I, 'DXXXXXXXX', 'YYYYYYYY')

onde:

I - Variável numérica de uma posição. O conteúdo que irá receber pode ser:

I=1 seta PASSWORD

I=2 apaga PASSWORD

I=3 habilita acesso no arquivo

I=4 desabilita acesso aos arquivos

D - Drive em que se encontra o programa que será acessado

XXXXXXXX - Nome do arquivo que será acessado, tamanho de 8 caracteres, caso o nome for menor, preencher com brancos à direita.

YYYYYYYY - Deve conter a senha do arquivo à ser acessado, abrangendo 8 caracteres, caso o nome seja menor que 8 caracteres, preencher com brancos à direita.

B.2. O código de erro será fornecido como retorno na variável I.

CÓDIGO	SIGNIFICADO
Ø	operação bem sucedida
1	parâmetro código inválido
2	erro no nome do arquivo
3	overflow na tabela de PASSWORD
4	erro de escrita na tabela
5	arquivo sem PASSWORD
6	PASSWORD inválido p/ arquivo
7	erro leitura na tabela PASSWORD
8	arquivo já existe na tabela PASSWORD.

Se o código de erro for = 4, utilizar a seguinte sintaxe:

CALL SENHA (I)

Não é necessário o nome do arquivo e a senha do arquivo

5.4 - EXEMPLOS:

A seguir temos dois exemplos de programas em FORTRAN:

A - IMPRFOR.FOR
B - SENHAFOR.FOR

que utilizam as rotinas SPOOL e SENHA respectivamente.

FILE: IMPRFOR.FOR

```
DIMENSION A1(10), J1(10)
K=Ø
CALL SPOOL (K)
N=K
IF (N) 15, 7, 15
7 DO 5 I=1,5
5 READ (1,10) A1(I), J1(I)
DO 12 J1=1,5
12 WRITE (2,11) A1(J), J1(J)
WRITE (2,25)
25 FORMAT (/)
10 FORMAT (A4, I2)
11 FORMAT (5X,A4,I2,/)
K=1
CALL SPOOL (K)
N=K
15 WRITE (1, 17) N
17 FORMAT (13)
END
```

FILE: SENHAFOR.FOR

```
      READ (1,10) I
      GO TO (35,2,11,19,30), I
2     WRITE(1,5)
5     FORMAT (1X,'DESTRUIR')
      GO TO 35
10    FORMAT (I2)
11    WRITE (1,15)
15    FORMAT(1X,'LIVRE ACESSO')
      GO TO 35
19    WRITE(1,20)
20    FORMAT(1X,'ACESSO INTERR.')
      GO TO 35
25    FORMAT(1X,'FIM')
35    CALL SENHA (I,'ATES3      ','PASSWORD')
      WRITE(1,31) I
31    FORMAT (I2)
30    WRITE (1,25)
      END
```

6 - COBOL MB

6.1 - DIFERENÇAS APRESENTADAS PELO COBOL MB

O COBOL da MICROBASE diferencia do COBOL 80 nos seguintes aspectos:

- Desempenho superior para método de acesso indexado
- Maior segurança e confiabilidade dos dados
- Arquivos compatíveis com outras linguagens
- Economia de espaço em disco para programas e arquivos indexados
- Economia de memória
- Linguagem voltada para Sistema multiusuário
- Arquivos indexados com múltiplas chaves
- Dois terminais podem utilizar o mesmo compilador ou linkeditor ao mesmo tempo, sendo gerados arquivos intermediários com nomes diferentes.

6.2 - ARQUIVOS

O COBOL MB admite funções de "LOCK" e "UNLOCK" para proteção a nível de registros. Esta implementação só é válida para programas desenvolvidos e compilados com o COBOL MB, em virtude do mesmo possuir a característica de multiusuário. Situações válidas para "LOCK" e "UNLOCK" de registros:

- a) A proteção a nível de registros só é permitida para arquivos de acesso relativo e indexado, abertos em I-O, sendo necessário que os arquivos estejam pré-formatados.
- b) Cada um dos terminais podem proteger até 10 (Dez) registros pertencentes à arquivos diferentes.
- c) Ao executar um comando "READ" para um determinado registro, este fica "LOCKED" ao terminal, não sendo possível o acesso a este registro por qualquer outro usuário, sendo liberado quando um novo registro for lido ou um comando CLOSE for executado para o arquivo.
- d) Antes da execução de um comando "REWRITE" é obrigatório um comando "READ", qualquer que seja o modo de acesso.
- e) Ao executar o comando "DELETE" para um registro, o COBOL MB protege o registro, deleta o mesmo e o desprotege. O próximo comando de leitura neste registro retornará um FILE STATUS = 23 (chave inexistente).
- f) Quando é executado um comando START, o registro a partir do qual a iniciada a leitura, ficará protegido. O registro só será liberado para o outro terminal, quando for lido o próximo registro ou executado um CLOSE no arquivo.

6.3 - BIBLIOTECA - 'LIVRONET'

Este item descreve as rotinas especiais que foram desenvolvidas para o POLY 910 NET para serem utilizadas com o COBOL MB. Essas rotinas estão embutidas em um único módulo, que deverá estar residente em disco na montagem.

Dependendo da configuração utilizada, este deverá ser "LIVRONET.REL" para Disco Rígido Cartridge ou "COBNETW.REL" para disco Winchester.

As chamadas dessas rotinas serão descritas a seguir, devendo para sua montagem:

- . Estar com o programa principal em código relocável (.REL) no disco.
- . Conter o programa LINK.COM e as rotinas COBLIB.REL e CRTDRV.REL no disco.
- . Conter o módulo LIVRONET.REL ou COBNETW.REL no disco utilizando o seguinte formato:

LINK/C, D:NOME1, D:NOME2, D:LIVRONET/S/A

ou

LINK/C, D:NOME1, D:NOME2, D:COBNETW/S/A ;onde:

D:NOME1 - é o drive e o nome do programa final em código executável (.COM)
D:NOME2 - é o drive e o nome do programa principal em código relocável
D:LIVRONET - drive e módulo LIVRONET.REL
D:COBNETW - drive e módulo COBNETW.REL
/C - indica que o programa está em COBOL
/S - realiza uma pesquisa das rotinas a serem "linkadas"
/A - link automático com as rotinas COBLIB.REL e CRTDRV.REL

Análogo ao COBOL 80, este apresenta as rotinas de Spool e Senha descritas no item 3.3, podendo-se tomar os mesmos exemplos como referência (item 3.4).

