PCS 3216
Sistemas de Programação

João José Neto

Aula 10 – Montadores de um passo e
Alguns aspectos de sua implementação

MONTADORES DE UM SÓ PASSO

Montadores de dois passos

- Os montadores de dois passos, como foi mencionado antes, dividem as tarefas de montagem dos programas simbólicos entre os dois passos em que são estruturados
  - Especializando cada um deles
    - Primeiro passo – coleta de informações sobre os símbolos
    - Segundo passo – geração de código objeto
  - Isso propicia a obtenção de programas potentes e eficientes.

Algumas Inconveniências

- Todavia, muitas vezes, deseja-se construir montadores para máquinas ou situações em que a dupla leitura física do texto simbólico é indesejável. Por exemplo:
  - Quando os periféricos de leitura/gravação disponíveis são muito lentos
  - Quando o montador é intensamente utilizado, ou seja, a operação de montagem de programas simbólicos é uma atividade muito intensa no sistema
  - Quando o sistema desempenha muitas atividades e está sobrecarregado.
  - Quando se deseja que a operação de montagem seja muito rápida, por exemplo, em uma escola, para atendimento muito frequente de atividades discentes.

Dois passos

- Em montadores de dois passos, esse problema costuma ser atenuado:
  - Memorizando uma cópia do texto simbólico (em disco, por exemplo) durante a sua leitura, no primeiro passo;
  - Posteriormente, para completar a operação de montagem, tal imagem do programa-fonte construída no primeiro passo é (re)lida, desta vez a partir do disco, pelo segundo passo do montador (só o fato de essa releitura ser feita a partir de um arquivo em disco e não de periféricos lentos já colabora significativamente para uma redução substancial do tempo gasto pelo montador na atividade de releitura do programa fonte).

Uso em máquinas sem memória de massa

- Há casos em que se usa, alternativamente, a opção de fazer a operação de montagem em um único passo:
  - Por imposição de decisões de projeto
  - Quando não há disponibilidade de um dispositivo de memória de massa (disco, por ex.) para esta tarefa
  - Quando não se dispõe de periféricos de entrada/saída rápida para a leitura/gravação do programa fonte ou do programa objeto.
  - Quando limitações ou exigências da aplicação inviabilizam o uso de outras soluções

Montadores de um só passo

- Para isto, são construídos os chamados *montadores de passo único*, que leem o programa fonte uma só vez.
- Há dois tipos principais de montadores de um só passo:
  - Montadores *load and go* (ou *assemble and go*), em que o código-objeto é gerado diretamente na memória, ficando disponível para ser executado imediatamente
  - Montadores *que geram código-objeto carregável*, a ser posteriormente introduzido na memória para execução. Nesta modalidade, o código-objeto assume a forma de um script numérico de preenchimento da memória, que determina como depositar na memória o código-objeto. Essa ordem reflete a exata sequência em que as referências à frente, na memória, foram sendo conhecidas pelo montador, durante o processo de montagem do programa-fonte.

Funcionalidade idêntica

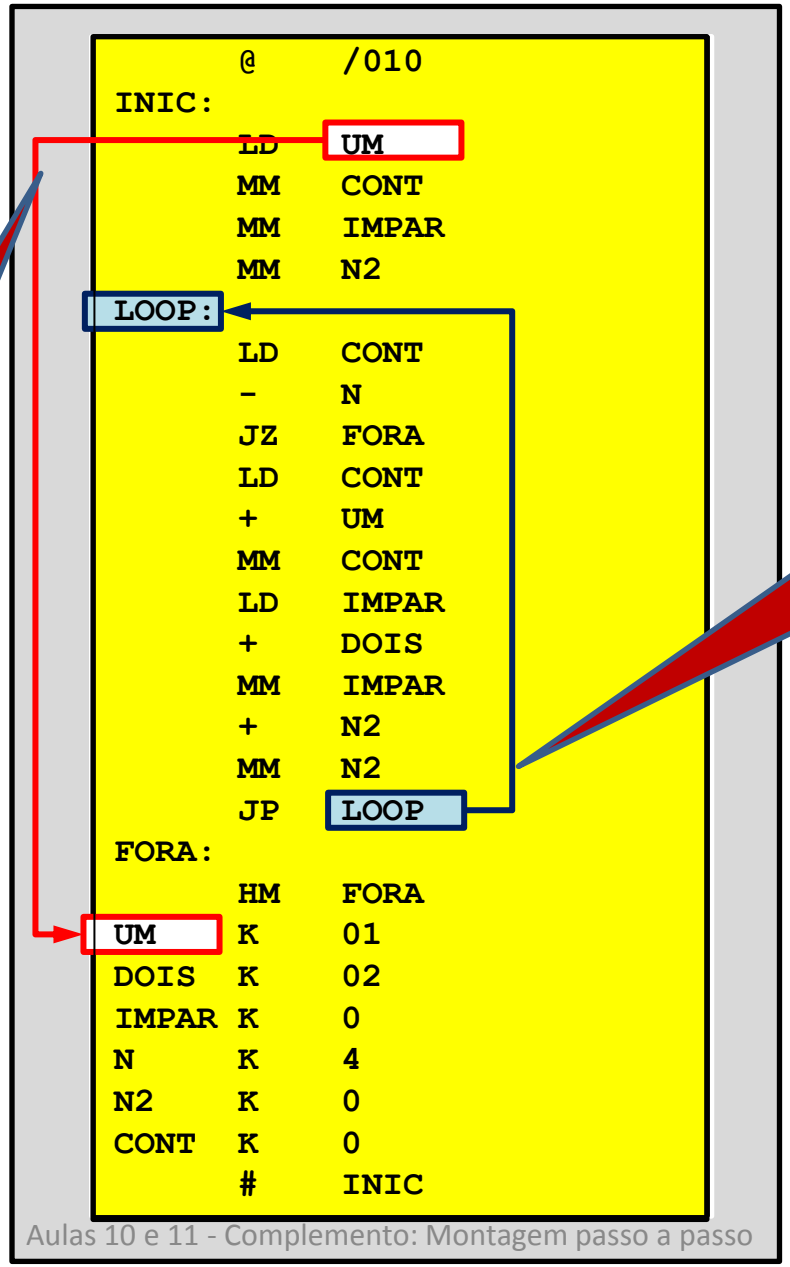
- Os montadores de um único passo devem executar funcionalmente todas as tarefas que foram descritas para os montadores de dois passos.
- Deverão fazê-lo, entretanto, sem exigir para isso mais de uma única leitura física do texto simbólico, por parte do montador.

Problema principal: Referências à frente

- Assim, será necessário resolver novamente alguns problemas que já foram solucionados pelo esquema em dois passos, os quais, porém, voltam à tona pela imposição de não seja feita mais que uma única operação de leitura física do programa-fonte simbólico.
- O problema mais sério a ser resolvido surge na ocasião da montagem de instruções de referência à memória, e decorre da falta de informação completa para a obtenção do código de máquina para essas instruções quando elas se apresentam com operandos simbólicos correspondentes a endereços ainda não definidos (referências simbólicas à frente).

Exemplos de referências à memória

Neste ponto o símbolo UM está indefinido, desconhecendo-se seu endereço, portanto.



Uma Referência para a frente

Uma Referência para trás

Já aqui o símbolo LOOP está definido, pois apareceu acima como rótulo, portanto desde aquele ponto seu endereço já estava determinado.

Uma Solução Prática

- Em montadores de um só passo, soluciona-se esta dificuldade através da memorização dessas instruções incompletas em uma tabela de códigos pendentes.
- Os elementos desta tabela referem-se ao conjunto de todas as instruções que apresentem operandos com endereços ainda não resolvidos, e devem registrar todas as informações conhecidas acerca dos respectivos operandos
- A finalidade desses registros é a de propiciar o preenchimento dessas informações assim que se tornarem disponíveis, durante a montagem do programa, ou seja, na ocasião da definição do símbolo do qual dependem.
- Assim, o código binário completo pode ser finalmente montado e gerado no programa-objeto, ou então, diretamente depositado na posição correta de memória.

O código-objeto a ser gerado

- Por questões práticas, o **código-objeto** a ser gerado pelo montador de um único passo deve ter **formato compatível** com os códigos-objeto adotados para o montador de dois passos, assim como para os demais programas do sistema de programação (loader, dumper, etc).
- Assim, um programa-objeto deverá constar de uma sequência de **um ou mais blocos** (binários ou hexadecimais), cada qual contendo os **códigos numéricos** correspondentes às instruções e dados do programa, acompanhados de **metadados** informando o **endereço** de memória associado a esses códigos, o **número de bytes** do bloco, e um byte de redundância contendo um "**checksum**", que costuma ser o byte mennos significativo da somatória de todos os bytes que compõem o bloco.
- O formato de cada bloco é o seguinte (= ao usado no caso de 2 passos):
  - Uma **separação física** (normalmente sequência de zeros binários)
  - Número N de **bytes do bloco** (maior que zero, obrigatoriamente)
  - **Endereço inicial** associado ao primeiro byte de código (n bytes para memórias com endereços de 2^n bits) No caso, 2 bytes (a memória tem 4096 bytes)
  - N bytes corespondentes ao **código numérico** contido no bloco
  - 1 byte de **checksum**, contendo o complemento da soma de todos os demais, de modo que seja nulo o último byte da soma de todos os bytes do bloco, incluindo o de checksum.

Geração do código-objeto

- Do ponto de vista de geração do código-objeto, é possível ao montador de um único passo **gerar códigos incompletos**, para as instruções com operandos não resolvidos, utilizando apenas as informações disponíveis na ocasião da leitura da instrução simbólica.
- A utilidade da geração desse código incompleto nessa ocasião é a de **reservar área de memória** para o código definitivo, cujo endereço e formato são conhecidos, mas cujo valor exato ainda não foi determinado.
- **Na ocasião da definição** do símbolo do qual dependem, as instruções cujos códigos ainda estejam inacabadas **podem ser completadas** com as informações faltantes, obtendo-se dessa forma o código final correspondente à instrução.

Lista de Pendências

- A **Lista de Pendências** costuma ser organizada como uma lista ligada, cujos elementos, além do **ponteiro para o próximo elemento** da lista, contêm no seu campo de informação triplas tais como (**EI, OPI, DI**), nas quais referenciam, respectivamente:
  - EI - **endereço de memória** associado à instrução correspondente
  - OPI - **código de operação** referente à instrução
  - DI - **deslocamento** (translação) a ser sofrido pelo endereço de definição do símbolo referenciado para formar corretamente o operando da instrução.

Estruturas de dados

- Possível implementação da lista de pendências
  - **Listas ligadas** implementam muito confortavelmente as listas de pendências, que representam o conjunto dos **códigos incompletos** presentes no programa-objeto em construção.
  - **A cada símbolo**, associa-se uma **lista de referências à frente ainda não resolvidas**, cada qual acompanhada de eventuais informações complementares sobre o operando associado.
  - Os elementos da lista de pendências são a ela **incorporados** toda vez que for encontrada uma **referência a um símbolo indefinido**, e são dela **removidos** sempre que o montador **determinar o endereço** a que se refira um desses símbolos.
  - Em algumas implementações, as listas de pendências são fisicamente armazenadas na mesma estrutura de dados da tabela de símbolos, funcionando assim como sua extensão.

Inclusão de pendências

- Na ocasião da inclusão de mais um elemento na lista de referências, pode-se verificar se o símbolo referenciado já está definido ou não.
- Se o símbolo não for encontrado, ou então, mesmo se for encontrado mas estiver indefinido, cria-se e inclui-se na lista de pendências associada a esse símbolo uma tripla da forma (EI, OPI, DI), conforme foi referido anteriormente. Nesse caso, nenhum código pode ser gerado nesta ocasião.
- Caso o símbolo já tenha sido encontrado, e se ele estiver marcado como já definido, então o respectivo endereço numérico já será conhecido, portanto nada de inédito haverá para ser memorizado, podendo portanto ser gerado diretamente o código definitivo para essa instrução, sem alterar a lista de pendências.

Resolução de pendências

- Posteriormente, na ocasião da definição do símbolo, a lista de pendências a ele associado deverá ter cada um dos seus elementos processado e eliminado, por meio da geração do código definitivo correspondente.
- Para isso constrói-se, para cada um dos elementos da lista, um bloco de código-objeto contendo as informações registradas na lista de pendências, complementadas com a informação do endereço de memória associado ao símbolo recém-definido.
- Após a geração de tal código, os elementos da lista de pendências referentes ao símbolo em questão podem ser descartados.

Geração descontínua de código

- Do ponto de vista da geração de código, se esta for efetuada diretamente na memória, não haverá qualquer necessidade de processamento adicional.
- Se, entretanto, a geração for feita em meio externo, como por exemplo em fita ou arquivo, certamente haverá uma descontinuidade na sequência de bytes gerados, em termos dos endereços de memória ocupados.
- Como consequência direta, o programa objeto resultante da montagem em passo único, ao contrário do que geralmente ocorre na montagem em dois passos, será composto de um número maior de blocos de código, e o comprimento desses blocos em geral será menor.

Esvaziamento do bloco de código

- Durante a geração de código, uma área de memória reservada a essa geração vai sendo preenchida até que o seu **comprimento máximo** seja atingido, ou até que se **altere o endereço de origem** do código.
- Neste último caso, deverá ser forçado o **esvaziamento** do bloco incompleto de código, em construção, e o **início do preenchimento** de um novo bloco.
- Isso deve ser feito para liberar espaço no bloco de código, cedendo área para a geração das informações de preenchimento de lacunas anteriormente criadas pelo montador, associadas a referências à frente representadas pelas pendências registradas.
- Obviamente, é possível usar uma lógica mais complexa para evitar a geração de alguns desses blocos adicionais, mas isso não será considerado no presente estudo.

Backtracking

- Esta técnica pode ser classificada na classe dos algoritmos de ***backtracking***, em que o trabalho não é efetuado linearmente, mas sofre **descontinuidades** para correções de códigos incompletos analisados anteriormente, e que estão portanto fora de ordem.
- O conteúdo do programa-objeto assim construído **reflete a ordem** na qual a memória é preenchida pelo montador à medida que este vai determinando os endereços associados aos diversos símbolos.
- Preserva, portanto, a ordem de preenchimento da memória, na sequência exata em que as informações forem sendo coletadas ou construídas **ao longo do trabalho de montagem**.

Ações de Retro-Preenchimento

- A execução das ações de "***backtracking***" costuma, portanto, envolver as seguintes ações:
  - **Salvamento do endereço** corrente de geração do código.
  - **Retroendereçamento** à posição ocupada pela instrução pendente que está sendo resolvida.
  - **Geração de bloco** de substituição (**preenchimento** ou **correção**) do código incompleto pela versão completa, agora conhecida, da instrução pendente.
  - **Abertura de novo bloco** de geração de código, restaurando a origem ao endereço inicialmente salvo, se for o caso (desnecessário se não houver alteração na origem do código gerado).

Interpretação

- Assim, o **código gerado** pelo montador de um passo não representa obrigatoriamente uma imagem do conteúdo da memória.
- Trata-se, na realidade, de um código que funciona como um "***script***" de preenchimento da memória.
- Naturalmente, esse programa de preenchimento é interpretado pelo ***loader*** (carregador) absoluto, o qual se encarrega de executar as ações apropriadas.
- Ao final dessa interpretação, os **bytes do código-objeto** representado nesse programa estará devidamente **carregado nos endereços corretos** da memória, pronto para a execução, se for o caso.

Listagens

- Um outro problema advindo da supressão de um passo na lógica do montador, manifesta-se nas **operações de listagem** do programa simbólico, ao lado do correspondente programa numérico.
- Optando-se por **efetuar a listagem à medida que se monta o programa**, surge um problema: havendo falta de informação, os códigos numéricos correspondentes às pendências não poderão ser listados na ocasião.
- Para salvar a listagem, pode-se **ao final da montagem imprimir a** tabela de endereços que tenham estado presentes na **lista de pendências** durante a montagem.
- Isso completa a informação impressa, mas o uso de uma listagem desse tipo continua **desconfortável**, pois registra mais o histórico da montagem do que o seu resultado final.

Listagem de códigos pendentes

- Muitos montadores de um passo imprimem os códigos gerados exatamente na mesma ordem em que os mesmos foram sendo gerados no programa objeto.
- Em outras palavras, os códigos pendentes são impressos logo após os símbolos dos quais dependem terem sido definidos (ocorrerem como rótulos).
- Outros montadores imprimem códigos incompletos, indicando que estão pendentes naquele ponto.
- Ao final da montagem, imprimem o conteúdo da tabela de símbolos, com cujo auxílio o usuário pode compor manualmente os endereços omitidos.

Listagens melhores

- As duas opções mencionadas fornecem listagens que se mostram incômodas para o usuário.
- Alguns montadores mais sofisticados, de um passo, geram em disco arquivos contendo uma imagem da listagem do programa e do código associado.
- Nesses arquivos, tais montadores alteram o conteúdo da imagem da listagem toda vez que for efetuada alguma resolução das referências à frente.
- Com esta última solução, é possível obter, com montadores de um passo, listagens idênticas às produzidas pelos de dois passos.

Desvantagem

- Como desvantagem principal do uso de montadores em um só passo, em geral é necessário ter à disposição uma considerável área de memória de massa.

- Incorporando as soluções apresentadas às ideias já utilizadas na lógica do montador de dois passos, é possível obter um montador capaz de efetuar em um único passo a montagem dos seus programas-fonte.

Conclusão

- Há naturalmente vantagens e desvantagens de se utilizar, na confecção de um montador, uma lógica de um ou de dois passos.
- Normalmente, deve-se levar em conta a finalidade a que o montador se destina, a frequência com que será utilizado, o tempo de retorno desejado, o tamanho médio dos programas que serão traduzidos, e o tamanho máximo dos mesmos, que irá servir para definir a dimensão de suas tabelas e áreas de dados, e mesmo, neste caso particular, o tipo mais adequado de montador.

EXEMPLO PASSO A PASSO DA OPERAÇÃO DE UM MONTADOR DE UM SÓ PASSO

Linha 1; CI antes=???; CI depois=/010

```
        @     /010
INIC:
        LD    UM
        MM    CONT
        MM    IMPAR
        MM    N2
LOOP:
        LD    CONT
        -     N
        JZ    FORA
        LD    CONT
        +     UM
        MM    CONT
        LD    IMPAR
        +     DOIS
        MM    IMPAR
        +     N2
        MM    N2
        JP    LOOP
FORA:
        HM    FORA
UM      K     01
DOIS    K     02
IMPAR   K     0
N       K     4
N2      K     0
CONT    K     0
        #     INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.

Códigos Pendentes

Saídas Geradas

Imagem simbólica do programa objeto

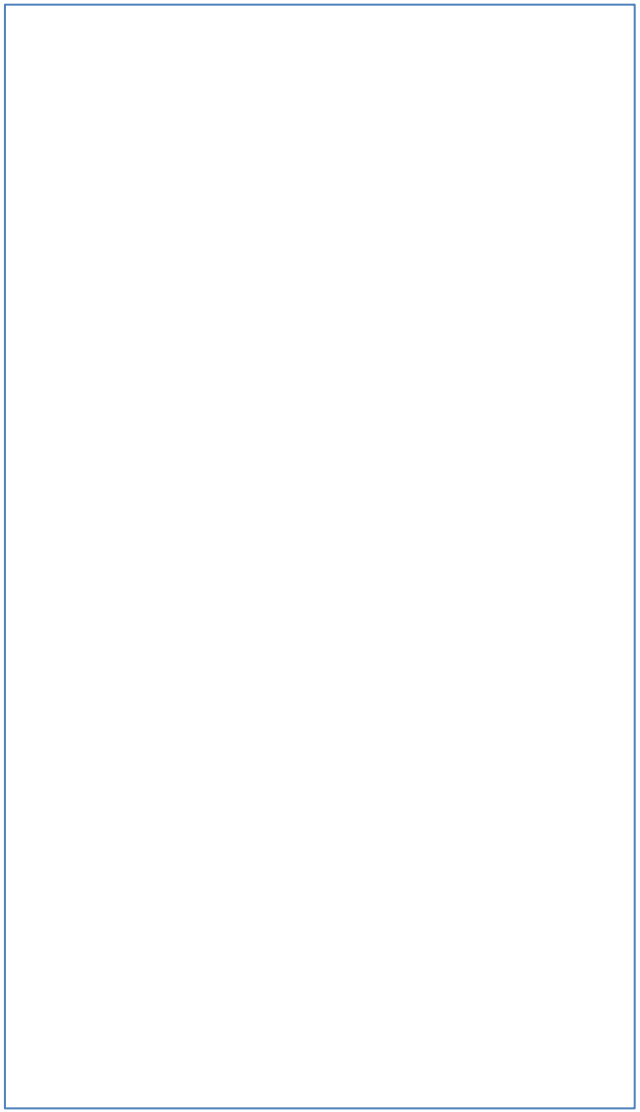
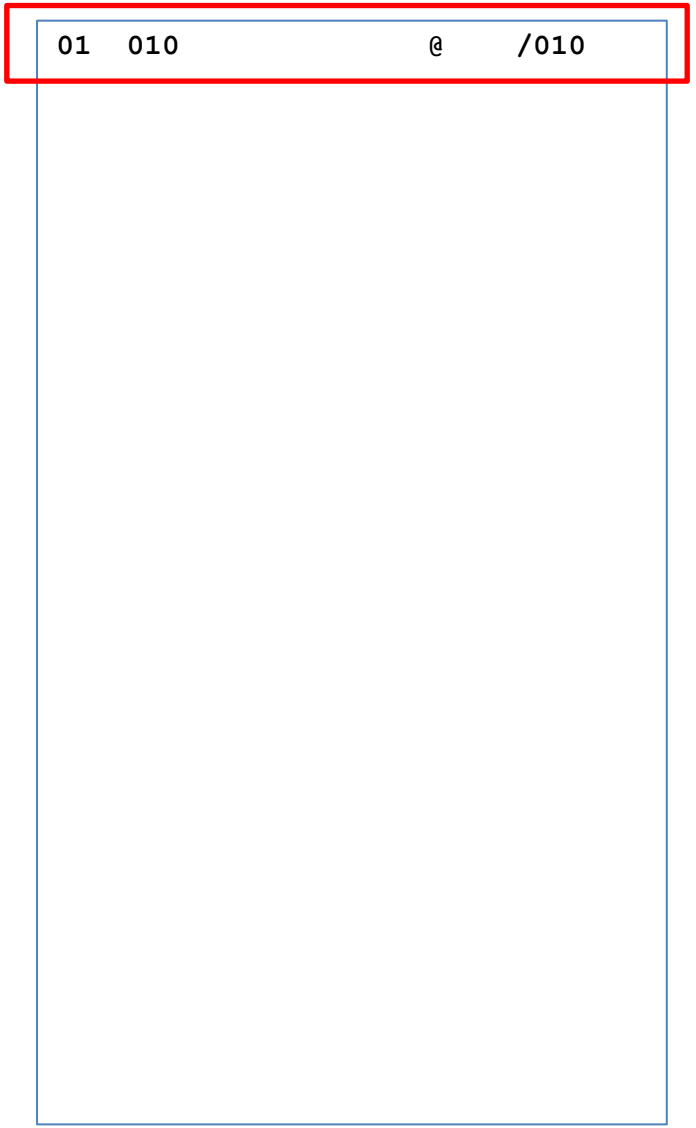


Imagem da Listagem



Linha 2; CI antes=/010; CI depois=/010

```
        @     /010
INIC:
        LD    UM
        MM    CONT
        MM    IMPAR
        MM    N2
LOOP:
        LD    CONT
        -     N
        JZ    FORA
        LD    CONT
        +     UM
        MM    CONT
        LD    IMPAR
        +     DOIS
        MM    IMPAR
        +     N2
        MM    N2
        JP    LOOP
FORA:
        HM    FORA
UM      K     01
DOIS    K     02
IMPAR   K     0
N       K     4
N2      K     0
CONT    K     0
        #     INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------

Códigos Pendentes

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010         INIC:
```

Linha 3; CI antes=/010; CI depois=/012

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
```

Linha 4; CI antes=/012; CI depois=/014

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
```

Linha 5; CI antes=/014; CI depois=/016

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
```

Linha 6; CI antes=/016; CI depois=/018

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
```

Linha 7; CI antes=/018; CI depois=/018

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018          LOOP:
```

Linha 8; CI antes=/018; CI depois=/01A

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018          LOOP:
08  018  8???           LD    CONT
```

Linha 9; CI antes=/01A; CI depois=/01C

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018          LOOP:
08  018  8???           LD    CONT
09  01A  5???           -     N
```

Linha 10; CI antes=/01C; CI depois=/01E

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018          LOOP:
08  018  8???           LD    CONT
09  01A  5???           -     N
10  01C  1???           JZ    FORA
```

Linha 11; CI antes=/01E; CI depois=/020

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018          LOOP:
08  018  8???           LD    CONT
09  01A  5???           -     N
10  01C  1???           JZ    FORA
11  01E  8???           LD    CONT
```

Linha 12; CI antes=/020; CI depois=/022

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	
020	4???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                    @     /010
02  010           INIC:
03  010  8???              LD    UM
04  012  9???              MM    CONT
05  014  9???              MM    IMPAR
06  016  9???              MM    N2
07  018           LOOP:
08  018  8???              LD    CONT
09  01A  5???              -     N
10  01C  1???              JZ    FORA
11  01E  8???              LD    CONT
12  020  4???              +     UM
```

Linha 13; CI antes=/022; CI depois=/024

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	
020	4???	
022	9???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018          LOOP:
08  018  8???           LD    CONT
09  01A  5???           -     N
10  01C  1???           JZ    FORA
11  01E  8???           LD    CONT
12  020  4???           +     UM
13  022  9???           MM    CONT
```

Linha 14; CI antes=/024; CI depois=/026

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	
020	4???	
022	9???	
024	8???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018          LOOP:
08  018  8???           LD    CONT
09  01A  5???           -     N
10  01C  1???           JZ    FORA
11  01E  8???           LD    CONT
12  020  4???           +     UM
13  022  9???           MM    CONT
14  024  8???           LD    IMPAR
```

Linha 15; CI antes=/026; CI depois=/028

```
        @     /010
INIC:
        LD    UM
        MM    CONT
        MM    IMPAR
        MM    N2
LOOP:
        LD    CONT
        -     N
        JZ    FORA
        LD    CONT
        +     UM
        MM    CONT
        LD    IMPAR
        +     DOIS
        MM    IMPAR
        +     N2
        MM    N2
        JP    LOOP
FORA:
        HM    FORA
UM      K     01
DOIS    K     02
IMPAR   K     0
N       K     4
N2      K     0
CONT    K     0
        #     INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		
DOIS	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	
020	4???	
022	9???	
024	8???	
026	4???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018          LOOP:
08  018  8???           LD    CONT
09  01A  5???           -     N
10  01C  1???           JZ    FORA
11  01E  8???           LD    CONT
12  020  4???           +     UM
13  022  9???           MM    CONT
14  024  8???           LD    IMPAR
15  026  4???           +     DOIS
```

Linha 16; CI antes=/028; CI depois=/02A

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		
DOIS	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	
020	4???	
022	9???	
024	8???	
026	4???	
028	9???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010           INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018           LOOP:
08  018  8???           LD    CONT
09  01A  5???           -     N
10  01C  1???           JZ    FORA
11  01E  8???           LD    CONT
12  020  4???           +     UM
13  022  9???           MM    CONT
14  024  8???           LD    IMPAR
15  026  4???           +     DOIS
16  028  9???           MM    IMPAR
```

Linha 17; CI antes=/02A; CI depois=/02C

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		
DOIS	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	
020	4???	
022	9???	
024	8???	
026	4???	
028	9???	
02A	4???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010           INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018           LOOP:
08  018  8???           LD    CONT
09  01A  5???           -     N
10  01C  1???           JZ    FORA
11  01E  8???           LD    CONT
12  020  4???           +     UM
13  022  9???           MM    CONT
14  024  8???           LD    IMPAR
15  026  4???           +     DOIS
16  028  9???           MM    IMPAR
17  02A  4???           +     N2
```

Linha 18; CI antes=/02C; CI depois=/02E

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		
DOIS	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	
020	4???	
022	9???	
024	8???	
026	4???	
028	9???	
02A	4???	
02C	9???	

Saídas Geradas

Imagem simbólica do programa objeto

Imagem da Listagem

```
01  010                 @     /010
02  010           INIC:
03  010  8???           LD    UM
04  012  9???           MM    CONT
05  014  9???           MM    IMPAR
06  016  9???           MM    N2
07  018           LOOP:
08  018  8???           LD    CONT
09  01A  5???           -     N
10  01C  1???           JZ    FORA
11  01E  8???           LD    CONT
12  020  4???           +     UM
13  022  9???           MM    CONT
14  024  8???           LD    IMPAR
15  026  4???           +     DOIS
16  028  9???           MM    IMPAR
17  02A  4???           +     N2
18  02C  9???           MM    N2
```

Linha 19; CI antes=/02E; CI depois=/030

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	IND		
DOIS	ROT	IND		

Código gerado (definitivo)

19	02E	0018	JP	LOOP

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	
020	4???	
022	9???	
024	8???	
026	4???	
028	9???	
02A	4???	
02C	9???	

Saídas Geradas

Imagem simbólica do programa objeto

```
19  02E  0018        JP   LOOP
```

Imagem da Listagem

```
01  010                  @    /010
02  010          INIC:
03  010  8???           LD   UM
04  012  9???           MM   CONT
05  014  9???           MM   IMPAR
06  016  9???           MM   N2
07  018          LOOP:
08  018  8???           LD   CONT
09  01A  5???           -    N
10  01C  1???           JZ   FORA
11  01E  8???           LD   CONT
12  020  4???           +    UM
13  022  9???           MM   CONT
14  024  8???           LD   IMPAR
15  026  4???           +    DOIS
16  028  9???           MM   IMPAR
17  02A  4???           +    N2
18  02C  9???           MM   N2
19  02E  0018           JP   LOOP
```

Linha 20; CI antes=/030; CI depois=/030

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	DEF	/030	
DOIS	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01C	1???	
01E	8???	
020	4???	
022	9???	
024	8???	
026	4???	
028	9???	
02A	4???	
02C	9???	

Código gerado (definitivo)

```
10   01C   1030        JZ    FORA
```

Saídas Geradas

Imagem simbólica do programa objeto

```
10   01C   1030        JZ    FORA

19   02E   0018        JP    LOOP
```

Imagem da Listagem

```
01   010                    @     /010
02   010           INIC:
03   010   8???             LD    UM
04   012   9???             MM    CONT
05   014   9???             MM    IMPAR
06   016   9???             MM    N2
07   018           LOOP:
08   018   8???             LD    CONT
09   01A   5???             -     N
10   01C   1030              JZ    FORA
11   01E   8???             LD    CONT
12   020   4???             +     UM
13   022   9???             MM    CONT
14   024   8???             LD    IMPAR
15   026   4???             +     DOIS
16   028   9???             MM    IMPAR
17   02A   4???             +     N2
18   02C   9???             MM    N2
19   02E   0018              JP    LOOP
20   030           FORA:
```

Linha 21; CI antes=/030; CI depois=/032

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	IND		
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	DEF	/030	
DOIS	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01E	8???	
020	4???	
022	9???	
024	8???	
026	4???	
028	9???	
02A	4???	
02C	9???	

Código gerado (definitivo)

21	030	C030	HM FORA

Saídas Geradas

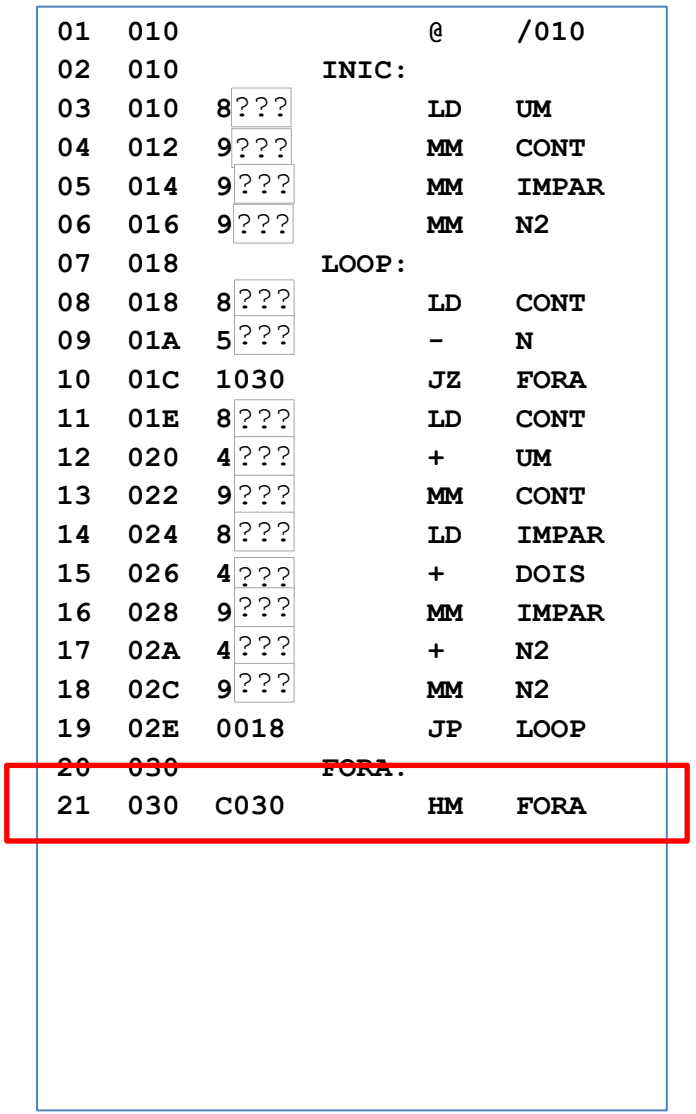
Imagem simbólica do programa objeto

```
10   01C   1030        JZ    FORA

19   02E   0018        JP    LOOP

21   030   C030        HM    FORA
```

Imagem da Listagem



```
01  010                  @    /010
02  010           INIC:
03  010  8???            LD   UM
04  012  9???            MM   CONT
05  014  9???            MM   IMPAR
06  016  9???            MM   N2
07  018           LOOP:
08  018  8???            LD   CONT
09  01A  5???            -    N
10  01C  1030            JZ   FORA
11  01E  8???            LD   CONT
12  020  4???            +    UM
13  022  9???            MM   CONT
14  024  8???            LD   IMPAR
15  026  4???            +    DOIS
16  028  9???            MM   IMPAR
17  02A  4???            +    N2
18  02C  9???            MM   N2
19  02E  0018            JP   LOOP
20  030           FORA:
21  030  C030            HM   FORA
```

Linha 22; CI antes=/032; CI depois=/033

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	DEF	/032	
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	DEF	/030	--------
DOIS	ROT	IND		

Códigos Pendentes

End.	Cód.	Pont.
010	8???	
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01E	8???	
020	4???	
022	9???	
024	8???	
026	4???	
028	9???	
02A	4???	
02C	9???	

Código gerado (definitivo)

```
22  032   01    UM    K     01
03  010   8032        LD    UM
12  020   4032        +     UM
```

Saídas Geradas

Imagem simbólica do programa objeto

```
03   010   8032          LD    UM

10   01C   1030          JZ    FORA

12   020   4032          +     UM

19   02E   0018          JP    LOOP

21   030   C030          HM    FORA
22   032   01      UM    K     01
```

Imagem da Listagem

```
01   010                      @     /010
02   010           INIC:
03   010   8032               LD    UM
04   012   9???               MM    CONT
05   014   9???               MM    IMPAR
06   016   9???               MM    N2
07   018           LOOP:
08   018   8???               LD    CONT
09   01A   5???               -     N
10   01C   1030               JZ    FORA
11   01E   8???               LD    CONT
12   020   4032               +     UM
13   022   9???               MM    CONT
14   024   8???               LD    IMPAR
15   026   4???               +     DOIS
16   028   9???               MM    IMPAR
17   02A   4???               +     N2
18   02C   9???               MM    N2
19   02E   0018               JP    LOOP
20   030           FORA:
21   030   C030               HM    FORA
22   032   01      UM         K     01
```

Linha 23; CI antes=/033; CI depois=/034

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	DEF	/032	--------
CONT	ROT	IND		
IMPAR	ROT	IND		
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	DEF	/030	--------
DOIS	ROT	DEF	/033	

Códigos Pendentes

End.	Cód.	Pont.
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01E	8???	
022	9???	
024	8???	
026	4???	
028	9???	
02A	4???	
02C	9???	

Código gerado (definitivo)

```
23  033  02    DOIS  K   02
15  026  4033        +   DOIS
```

Saídas Geradas

Imagem simbólica do programa objeto

```
03   010   8032          LD    UM

10   01C   1030          JZ    FORA

12   020   4032          +     UM

15   026   4033          +     DOIS

19   02E   0018          JP    LOOP

21   030   C030          HM    FORA
22   032   01      UM    K     01
23   033   02      DOIS  K     02
```

Imagem da Listagem

```
01   010                 @     /010
02   010           INIC:
03   010   8032          LD    UM
04   012   9???          MM    CONT
05   014   9???          MM    IMPAR
06   016   9???          MM    N2
07   018           LOOP:
08   018   8???          LD    CONT
09   01A   5???          -     N
10   01C   1030          JZ    FORA
11   01E   8???          LD    CONT
12   020   4032          +     UM
13   022   9???          MM    CONT
14   024   8???          LD    IMPAR
15   026   4033          +     DOIS
16   028   9???          MM    IMPAR
17   02A   4???          +     N2
18   02C   9???          MM    N2
19   02E   0018          JP    LOOP
20   030           FORA:
21   030   C030          HM    FORA
22   032   01      UM    K     01
23   033   02      DOIS  K     02
```

Linha 24; CI antes=/034; CI depois=/035

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	DEF	/032	--------
CONT	ROT	IND		
IMPAR	ROT	DEF	/034	
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	IND		
FORA	ROT	DEF	/030	--------
DOIS	ROT	DEF	/033	--------

Códigos Pendentes

End.	Cód.	Pont.
012	9???	
014	9???	
016	9???	
018	8???	
01A	5???	
01E	8???	
022	9???	
024	8???	
028	9???	
02A	4???	
02C	9???	

Código gerado (definitivo)

```
24  034  00   IMPAR   K   0
05  014  9034   MM   IMPAR
14  024  8034   LD   IMPAR
16  028  9034   MM   IMPAR
```

Saídas Geradas

Imagem simbólica do programa objeto

```
03  010  8032          LD   UM

05  014  9034          MM   IMPAR




10  01C  1030          JZ   FORA

12  020  4032          +    UM

14  024  8034          LD   IMPAR
15  026  4033          +    DOIS
16  028  9034          MM   IMPAR


19  02E  0018          JP   LOOP

21  030  C030          HM   FORA
22  032  01    UM      K    01
23  033  02    DOIS    K    02
24  034  00    IMPAR   K    0
```

Imagem da Listagem

```
01  010                 @    /010
02  010           INIC:
03  010  8032           LD   UM
04  012  9???           MM   CONT
05  014  9034           MM   IMPAR
06  016  9???           MM   N2
07  018           LOOP:
08  018  8???           LD   CONT
09  01A  5???           -    N
10  01C  1030           JZ   FORA
11  01E  8???           LD   CONT
12  020  4032           +    UM
13  022  9???           MM   CONT
14  024  8034           LD   IMPAR
15  026  4033           +    DOIS
16  028  9034           MM   IMPAR
17  02A  4???           +    N2
18  02C  9???           MM   N2
19  02E  0018           JP   LOOP
20  030           FORA:
21  030  C030           HM   FORA
22  032  01    UM      K    01
23  033  02    DOIS    K    02
24  034  00    IMPAR   K    0
```

Linha 25; CI antes=/035; CI depois=/036

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	DEF	/032	--------
CONT	ROT	IND		
IMPAR	ROT	DEF	/034	--------
N2	ROT	IND		
LOOP	ROT	DEF	/018	--------
N	ROT	DEF	/035	
FORA	ROT	DEF	/030	--------
DOIS	ROT	DEF	/033	--------

Códigos Pendentes

End.	Cód.	Pont.
012	9???	
016	9???	
018	8???	
01A	5???	
01E	8???	
022	9???	
02A	4???	
02C	9???	

Código gerado (definitivo)

```
25   035   04    N     K     4
09   01A   5035        -     N
```

Saídas Geradas

Imagem simbólica do programa objeto

```
03  010  8032          LD   UM

05  014  9034          MM   IMPAR



09  01A  5035          -    N
10  01C  1030          JZ   FORA

12  020  4032          +    UM

14  024  8034          LD   IMPAR
15  026  4033          +    DOIS
16  028  9034          MM   IMPAR


19  02E  0018          JP   LOOP

21  030  C030          HM   FORA
22  032  01    UM      K    01
23  033  02    DOIS    K    02
24  034  00    IMPAR   K    0
25  035  04    N       K    4
```

Imagem da Listagem

```
01  010                 @    /010
02  010           INIC:
03  010  8032           LD   UM
04  012  9???           MM   CONT
05  014  9034           MM   IMPAR
06  016  9???           MM   N2
07  018           LOOP:
08  018  8???           LD   CONT
09  01A  5035           -    N
10  01C  1030           JZ   FORA
11  01E  8???           LD   CONT
12  020  4032           +    UM
13  022  9???           MM   CONT
14  024  8034           LD   IMPAR
15  026  4033           +    DOIS
16  028  9034           MM   IMPAR
17  02A  4???           +    N2
18  02C  9???           MM   N2
19  02E  0018           JP   LOOP
20  030           FORA:
21  030  C030           HM   FORA
22  032  01    UM       K    01
23  033  02    DOIS     K    02
24  034  00    IMPAR    K    0
25  035  04    N        K    4
```

Linha 26; CI antes=/036; CI depois=/037

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	DEF	/032	--------
CONT	ROT	IND		
IMPAR	ROT	DEF	/024	--------
N2	ROT	DEF	/036	
LOOP	ROT	DEF	/018	--------
N	ROT	DEF	035	--------
FORA	ROT	DEF	/030	--------
DOIS	ROT	DEF	/033	--------

Códigos Pendentes

End.	Cód.	Pont.
012	9???	
016	9???	
018	8???	
01E	8???	
022	9???	
02A	4???	
02C	9???	

Código gerado (definitivo)

```
26  036  00    N2      K     0
06  016  9036          MM    N2
17  02A  4036          +     N2
18  02C  9036          MM    N2
```

Saídas Geradas

Imagem simbólica do programa objeto

```
03  010  8032          LD   UM

05  014  9034          MM   IMPAR
06  016  9036          MM   N2


09  01A  5035          -    N
10  01C  1030          JZ   FORA

12  020  4032          +    UM

14  024  8034          LD   IMPAR
15  026  4033          +    DOIS
16  028  9034          MM   IMPAR
17  02A  4036          +    N2
18  02C  9036          MM   N2
19  02E  0018          JP   LOOP

21  030  C030          HM   FORA
22  032  01    UM      K    01
23  033  02    DOIS    K    02
24  034  00    IMPAR   K    0
25  035  04    N       K    4
26  036  00    N2      K    0
```

Imagem da Listagem

```
01  010                 @    /010
02  010           INIC:
03  010  8032           LD   UM
04  012  9???           MM   CONT
05  014  9034           MM   IMPAR
06  016  9036           MM   N2
07  018           LOOP:
08  018  8???           LD   CONT
09  01A  5035           -    N
10  01C  1030           JZ   FORA
11  01E  8???           LD   CONT
12  020  4032           +    UM
13  022  9???           MM   CONT
14  024  8034           LD   IMPAR
15  026  4033           +    DOIS
16  028  9034           MM   IMPAR
17  02A  4036           +    N2
18  02C  9036           MM   N2
19  02E  0018           JP   LOOP
20  030           FORA:
21  030  C030           HM   FORA
22  032  01    UM       K    01
23  033  02    DOIS     K    02
24  034  00    IMPAR    K    0
25  035  04    N        K    4
26  036  00    N2       K    0
```

Linha 27; CI antes=/037; CI depois=/038

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	DEF	/012	--------
CONT	ROT	IND	/037	
IMPAR	ROT	DEF	/034	--------
N2	ROT	DEF	/036	--------
LOOP	ROT	DEF	/018	--------
N	ROT	DEF	/035	--------
FORA	ROT	DEF	/030	--------
DOIS	ROT	DEF	/033	--------

Códigos Pendentes

End.	Cód.	Pont.
012	9???	
018	8???	
01E	8???	
022	9???	

Código gerado (definitivo)

```
27  037  00    CONT  K     0
04  012  9037        MM    CONT
08  018  8037        LD    CONT
11  01E  8037        LD    CONT
13  022  9037        MM    CONT
```

Saídas Geradas

Imagem simbólica do programa objeto

```
03  010  8032          LD    UM
04  012  9037          MM    CONT
05  014  9034          MM    IMPAR
06  016  9036          MM    N2

08  018  8037          LD    CONT
09  01A  5035          -     N
10  01C  1030          JZ    FORA
11  01E  8037          LD    CONT
12  020  4032          +     UM
13  022  9037          MM    CONT
14  024  8034          LD    IMPAR
15  026  4033          +     DOIS
16  028  9034          MM    IMPAR
17  02A  4036          +     N2
18  02C  9036          MM    N2
19  02E  0018          JP    LOOP

21  030  C030          HM    FORA
22  032  01    UM      K     01
23  033  02    DOIS    K     02
24  034  00    IMPAR   K     0
25  035  04    N       K     4
26  036  00    N2      K     0
27  037  00    CONT    K     0
```

Imagem da Listagem

```
01  010                 @     /010
02  010          INIC:
03  010  8032          LD    UM
04  012  9037          MM    CONT
05  014  9034          MM    IMPAR
06  016  9036          MM    N2
07  018          LOOP:
08  018  8037          LD    CONT
09  01A  5035          -     N
10  01C  1030          JZ    FORA
11  01E  8037          LD    CONT
12  020  4032          +     UM
13  022  9037          MM    CONT
14  024  8034          LD    IMPAR
15  026  4033          +     DOIS
16  028  9034          MM    IMPAR
17  02A  4036          +     N2
18  02C  9036          MM    N2
19  02E  0018          JP    LOOP
20  030          FORA:
21  030  C030          HM    FORA
22  032  01    UM      K     01
23  033  02    DOIS    K     02
24  034  00    IMPAR   K     0
25  035  04    N       K     4
26  036  00    N2      K     0
27  037  00    CONT    K     0
```

Linha 27; CI antes=/038; CI depois=/038

```
        @       /010
INIC:
        LD      UM
        MM      CONT
        MM      IMPAR
        MM      N2
LOOP:
        LD      CONT
        -       N
        JZ      FORA
        LD      CONT
        +       UM
        MM      CONT
        LD      IMPAR
        +       DOIS
        MM      IMPAR
        +       N2
        MM      N2
        JP      LOOP
FORA:
        HM      FORA
UM      K       01
DOIS    K       02
IMPAR   K       0
N       K       4
N2      K       0
CONT    K       0
        #       INIC
```

Tabela de Símbolos

Símb.	Tipo	Def.	End.	Pend.
INIC	ROT	DEF	/010	--------
UM	ROT	DEF	/032	--------
CONT	ROT	DEF	/037	--------
IMPAR	ROT	DEF	/034	--------
N2	ROT	DEF	/036	--------
LOOP	ROT	DEF	/018	--------
N	ROT	DEF	/035	--------
FORA	ROT	DEF	/030	--------
DOIS	ROT	DEF	/033	--------

Códigos Pendentes

End.	Cód.	Pont.

SITUAÇÃO FINAL

Listagem final do Programa montado

Coletando todos os códigos gerados, temos o programa totalmente montado abaixo:

```
01  010              @    /010
02  010       INIC:
03  010  8032       LD   UM
04  012  9037       MM   CONT
05  014  9034       MM   IMPAR
06  016  9036       MM   N2
07  018       LOOP:
08  018  8037       LD   CONT
09  01A  5035       -    N
10  01C  1030       JZ   FORA
11  01E  8037       LD   CONT
12  020  4032       +    UM
13  022  9037       MM   CONT
14  024  8034       LD   IMPAR
15  026  4033       +    DOIS
16  028  9034       MM   IMPAR
17  02A  4036       +    N2
18  02C  9036       MM   N2
19  02E  0018       JP   LOOP
20  030       FORA:
21  030  C030       HM   FORA
22  032  01    UM   K    01
23  033  02    DOIS K    02
24  034  00    IMPAR K   0
25  035  04    N    K    4
26  036  00    N2   K    0
27  037  00    CONT K    0
28  010              #    INIC
```

Código objeto, na sequência da geração

```
19  02E  0018          JP    LOOP

10  01C  1030          JZ    FORA

21  030  C030          HM    FORA

22  032  01     UM     K     01
03  010  8032          LD    UM
12  020  4032          +     UM

23  033  02     DOIS   K     02
15  026  4033          +     DOIS

24  034  00     IMPAR  K     0
05  014  9034          MM    IMPAR
14  024  8034          LD    IMPAR
16  028  9034          MM    IMPAR

25  035  04     N      K     4
09  01A  5035          -     N

26  036  00     N2     K     0
06  016  9036          MM    N2
17  02A  4036          +     N2
18  02C  9036          MM    N2

27  037  00     CONT   K     0
04  012  9037          MM    CONT
08  018  8037          LD    CONT
11  01E  8037          LD    CONT
13  022  9037          MM    CONT
```

```
          @     /02E
          JP    LOOP
```

05 00 2E 00 18 B5

```
          @     /01C
          JZ    FORA
```

05 00 1C 10 30 9F

```
          @     /030
          HM    FORA
UM        K     01
```

06 00 30 C0 30 01 D9

Código objeto, na sequência da geração

```
19  02E  0018          JP   LOOP

10  01C  1030          JZ   FORA

21  030  C030          HM   FORA

22  032  01     UM     K    01
03  010  8032          LD   UM
12  020  4032          +    UM

23  033  02     DOIS   K    02
15  026  4033          +    DOIS

24  034  00     IMPAR  K    0
05  014  9034          MM   IMPAR
14  024  8034          LD   IMPAR
16  028  9034          MM   IMPAR

25  035  04     N      K    4
09  01A  5035          -    N

26  036  00     N2     K    0
06  016  9036          MM   N2
17  02A  4036          +    N2
18  02C  9036          MM   N2

27  037  00     CONT   K    0
04  012  9037          MM   CONT
08  018  8037          LD   CONT
11  01E  8037          LD   CONT
13  022  9037          MM   CONT
```

```
        @     /010
        LD    UM
```

05 00 10 80 32 39

```
        @     /020
        +     UM
```

05 00 20 40 32 69

```
        @     /033
DOIS    K     02
```

04 00 33 02 C7

Código objeto, na sequência da geração

```
19  02E  0018          JP    LOOP

10  01C  1030          JZ    FORA

21  030  C030          HM    FORA

22  032  01    UM     K     01
03  010  8032          LD    UM
12  020  4032          +     UM

23  033  02    DOIS   K     02
15  026  4033          +     DOIS

24  034  00    IMPAR  K     0
05  014  9034          MM    IMPAR
14  024  8034          LD    IMPAR
16  028  9034          MM    IMPAR

25  035  04    N      K     4
09  01A  5035          -     N

26  036  00    N2     K     0
06  016  9036          MM    N2
17  02A  4036          +     N2
18  02C  9036          MM    N2

27  037  00    CONT   K     0
04  012  9037          MM    CONT
08  018  8037          LD    CONT
11  01E  8037          LD    CONT
13  022  9037          MM    CONT
```

```
        @     /026
        +     DOIS
```

05 00 26 40 33 CD

```
        @     /034
IMPAR   K     0
```

04 00 34 00 C8

```
        @     /014
        MM    IMPAR
```

05 00 14 90 34 23

Código objeto, na sequência da geração

```
19  02E  0018          JP    LOOP

10  01C  1030          JZ    FORA

21  030  C030          HM    FORA

22  032  01    UM     K     01
03  010  8032          LD    UM
12  020  4032          +     UM

23  033  02    DOIS   K     02
15  026  4033          +     DOIS

24  034  00    IMPAR  K     0
05  014  9034          MM    IMPAR
14  024  8034          LD    IMPAR
16  028  9034          MM    IMPAR

25  035  04    N      K     4
09  01A  5035          -     N

26  036  00    N2     K     0
06  016  9036          MM    N2
17  02A  4036          +     N2
18  02C  9036          MM    N2

27  037  00    CONT   K     0
04  012  9037          MM    CONT
08  018  8037          LD    CONT
11  01E  8037          LD    CONT
13  022  9037          MM    CONT
```

```
          @     /024
          LD    IMPAR
```

05 00 24 80 34 23

```
          @     /028
          MM    IMPAR
```

05 00 28 90 34 0F

```
          @     /014
N         K     4
```

04 00 35 04 C3

Código objeto, na sequência da geração

```
19  02E  0018        JP   LOOP

10  01C  1030        JZ   FORA

21  030  C030        HM   FORA

22  032  01    UM    K    01
03  010  8032        LD   UM
12  020  4032        +    UM

23  033  02    DOIS  K    02
15  026  4033        +    DOIS

24  034  00    IMPAR K    0
05  014  9034        MM   IMPAR
14  024  8034        LD   IMPAR
16  028  9034        MM   IMPAR

25  035  04    N     K    4
09  01A  5035        -    N

26  036  00    N2    K    0
06  016  9036        MM   N2
17  02A  4036        +    N2
18  02C  9036        MM   N2

27  037  00    CONT  K    0
04  012  9037        MM   CONT
08  018  8037        LD   CONT
11  01E  8037        LD   CONT
13  022  9037        MM   CONT
```

```
        @    /01A
        -    N
```

05 00 1A 50 35 5C

```
        @    /036
N2      K    0
```

04 00 36 00 C6

```
        @    /016
        MM   N2
```

05 00 16 90 36 1F

Código objeto, na sequência da geração

```
19  02E  0018        JP   LOOP

10  01C  1030        JZ   FORA

21  030  C030        HM   FORA

22  032  01    UM    K    01
03  010  8032        LD   UM
12  020  4032        +    UM

23  033  02    DOIS  K    02
15  026  4033        +    DOIS

24  034  00    IMPAR K    0
05  014  9034        MM   IMPAR
14  024  8034        LD   IMPAR
16  028  9034        MM   IMPAR

25  035  04    N     K    4
09  01A  5035        -    N

26  036  00    N2    K    0
06  016  9036        MM   N2
17  02A  4036        +    N2
18  02C  9036        MM   N2

27  037  00    CONT  K    0
04  012  9037        MM   CONT
08  018  8037        LD   CONT
11  01E  8037        LD   CONT
13  022  9037        MM   CONT
```

```
        @    /02A
        +    N2
        MM   N2
```

07 00 2A 40 36 90 36 93

```
        @    /037
CONT    K    0
```

04 00 37 00 C5

```
        @    /012
        MM   CONT
```

05 00 12 90 37 22

Código objeto, na sequência da geração

```
19  02E  0018          JP   LOOP

10  01C  1030          JZ   FORA

21  030  C030          HM   FORA

22  032  01    UM     K    01
03  010  8032          LD   UM
12  020  4032          +    UM

23  033  02    DOIS   K    02
15  026  4033          +    DOIS

24  034  00    IMPAR  K    0
05  014  9034          MM   IMPAR
14  024  8034          LD   IMPAR
16  028  9034          MM   IMPAR

25  035  04    N      K    4
09  01A  5035          -    N

26  036  00    N2     K    0
06  016  9036          MM   N2
17  02A  4036          +    N2
18  02C  9036          MM   N2

27  037  00    CONT   K    0
04  012  9037          MM   CONT
08  018  8037          LD   CONT
11  01E  8037          LD   CONT
13  022  9037          MM   CONT
```

```
@     /018
LD    CONT
```

05 00 18 80 37 2C

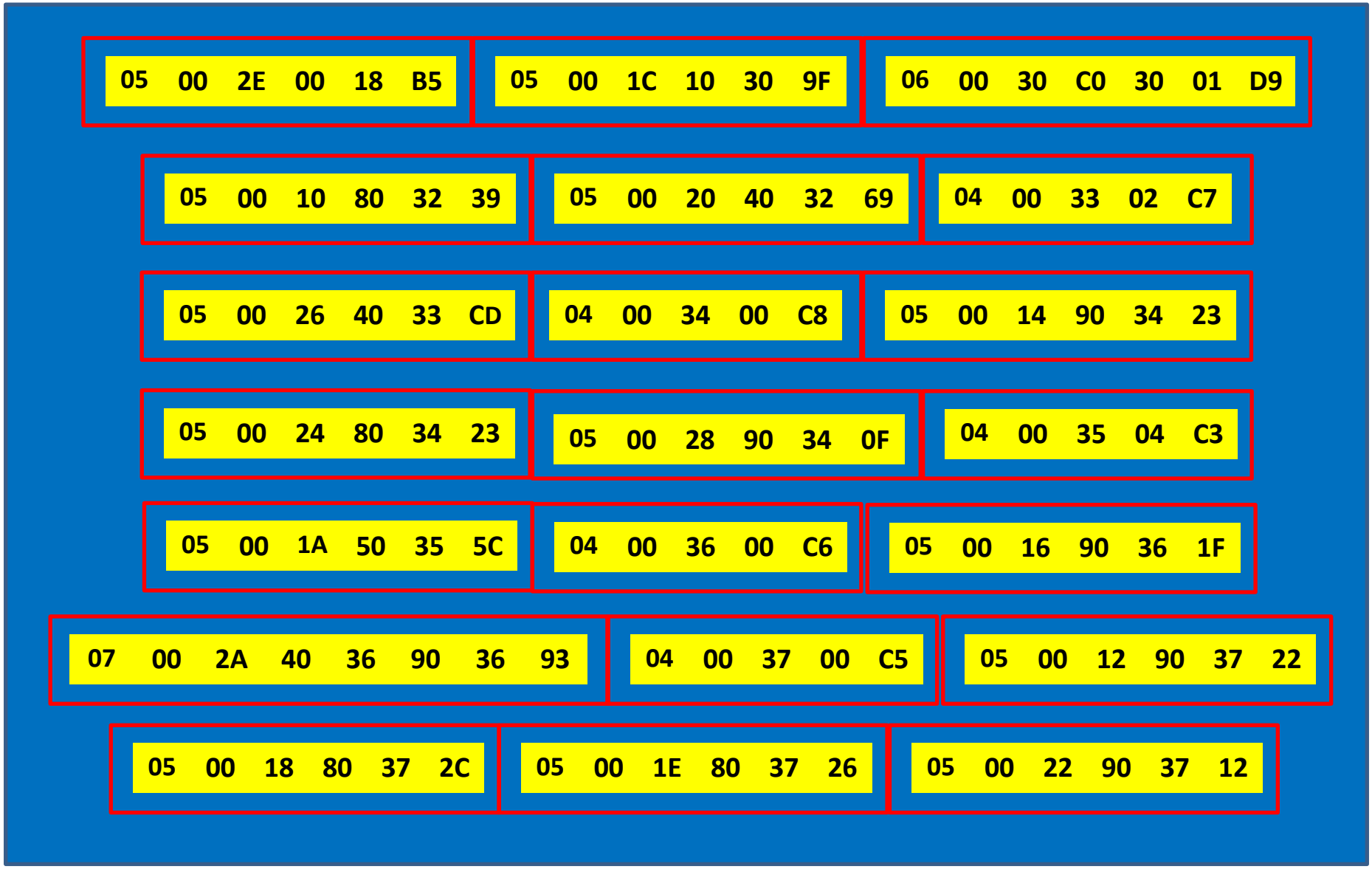
```
@     /01E
LD    CONT
```

05 00 1E 80 37 26

```
@     /022
MM    CONT
```

05 00 22 90 37 12

Cód. Obj. gerado pelo montador de 1 passo



Código objeto gerado pelo montador de 2 passos

É fácil verificar que este código, gerado pelo montador de dois passos, é equivalente ao anteriormente produzido pelo montador de um só passo.
No entanto, este se mostra mais compacto e mais fácil de ser compreendido, pois se mostra linear, sem descontinuidades e sem alterações de endereçamento.

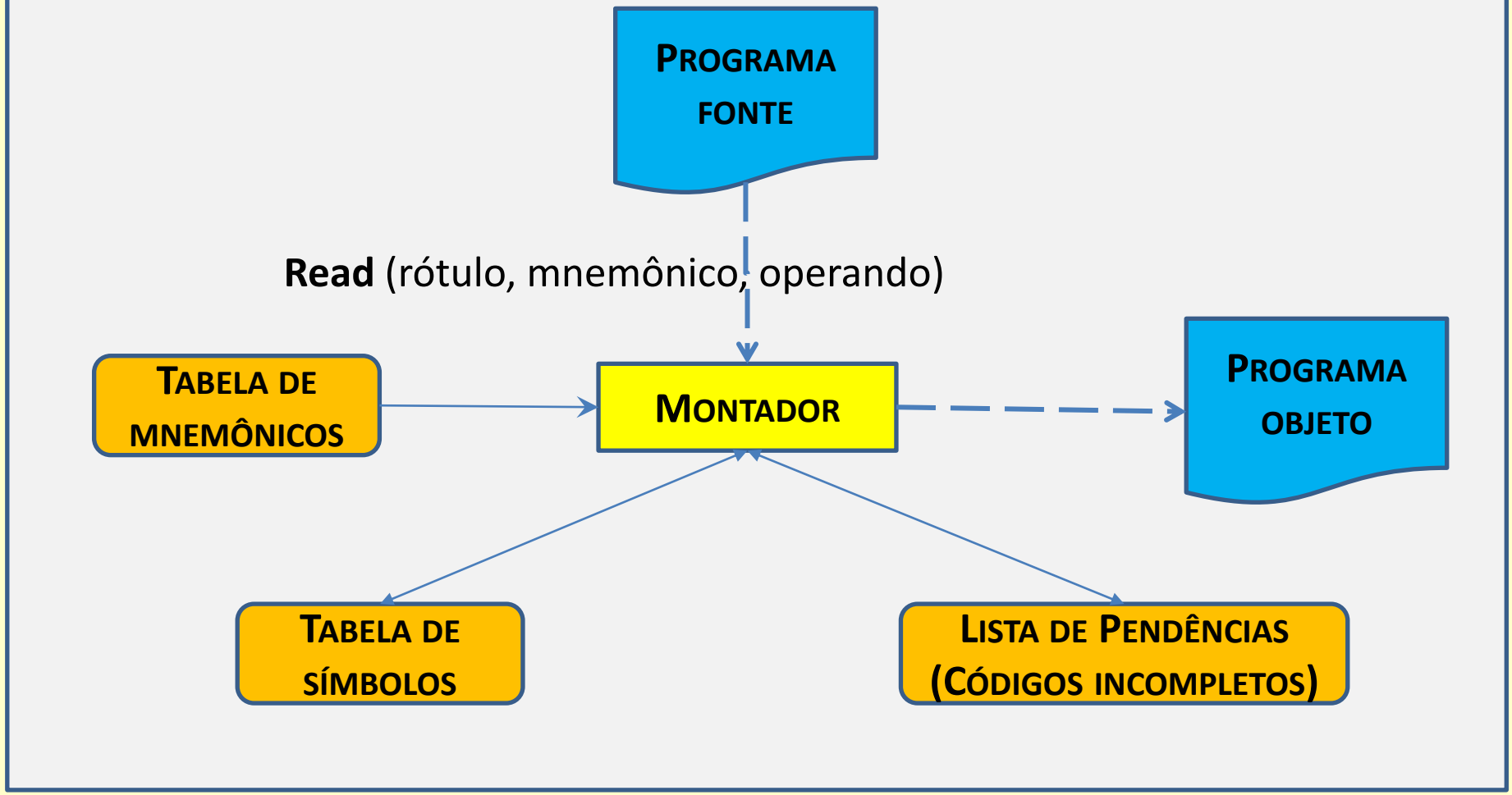
```
11 00 10 80 32 90 37 90 34 90 36 80 37 50 35 10 30 A0
```

```
11 00 1E 80 37 40 32 90 37 80 34 40 33 90 34 40 36 80
```

```
0F 00 2C 90 36 00 18 C0 30 01 02 00 04 00 00 10
```

ORGANIZAÇÃO DE UM MONTADOR DE UM ÚNICO PASSO

Organização de um montador simples, em um único passo



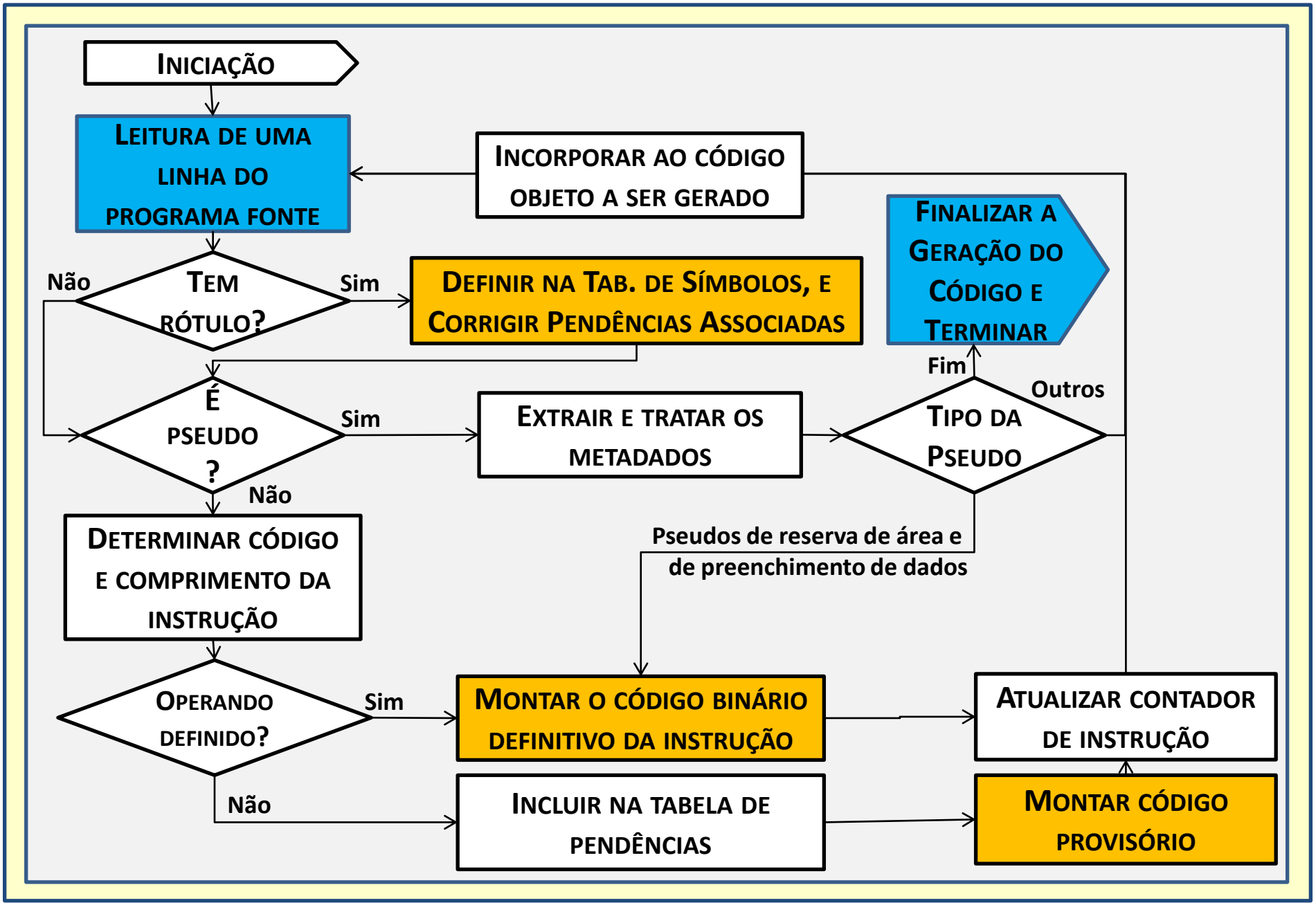
Implementação em passo único

- A organização em um passo apenas fornece uma arquitetura alternativa para a construção de um montador.
- Como nos montadores de mais de um passo as responsabilidades pela execução das tarefas é dividida, cada passo poderá ser implementado usando uma ***área menor de armazenamento.***
- Nos montadores de um passo, a ***leitura*** das entradas é feita uma só vez.
- Não havendo divisão de tarefas, a estrutura desse tipo de montador fica ***mais complexa***, pois precisa levar em conta diversos itens que a implementação em dois passos dispensa.
  - A **montagem da tabela de símbolos** é idêntica à do montador de dois passos
  - A **busca de inconsistências** na tabela de símbolos é feita ao final da montagem, quando o código objeto já está gerado
  - O **código-objeto** vai sendo montado ao longo da leitura do programa fonte, e pode ser absoluto ou relocável
  - O programa objeto, quando absoluto, pode ser gerado diretamente na memória ou **em formato carregável**

Atividades principais:

- Construção da tabela de símbolos
- Consulta à tabela de mnemônicos
- Consulta à tabela de códigos
- Construção da tabela de equivalências
- Determinação do endereço (absoluto ou relocável) para cada instrução
- Avaliação das expressões dos operandos
- Montagem do código objeto
- Teste de consistência da tabela de símbolos
- Geração de tabelas de referências cruzadas
- Geração da listagem formatada
- Geração do programa objeto

Lógica resumida do do montador de 1 passo



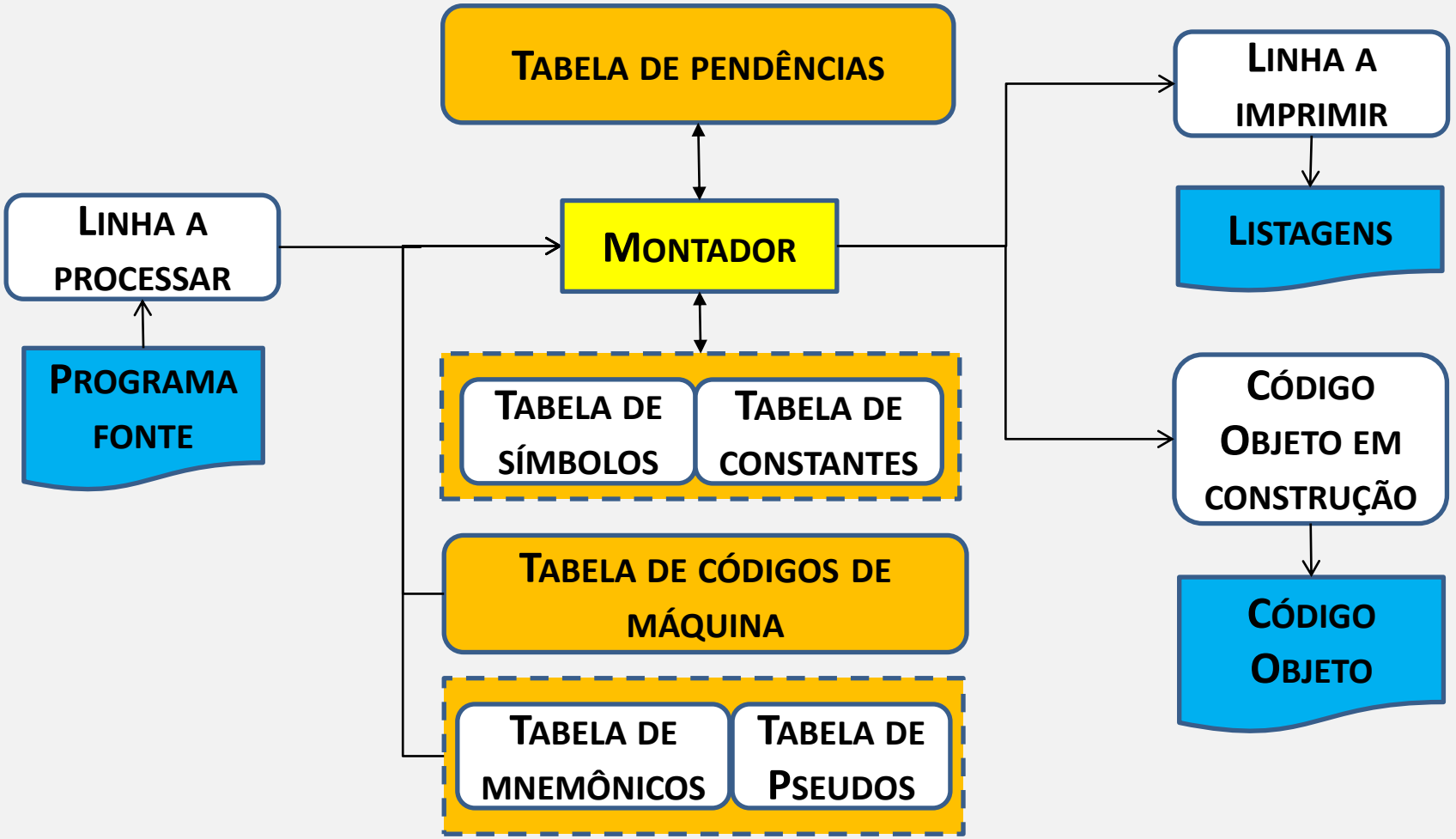
Principais Estruturas de Dados

- **Tabela de símbolos** (símbolo - endereço - definido - referenciado)
- Extensão da tabela de símbolo para geração de **referências cruzadas**
  - Linha de definição
  - Link para ordem alfabética
  - Ponteiro para lista de referências
  - Ponteiro para lista de pendências
- **Lista de referências** (para referências cruzadas)
  - Link
  - Número da Linha
- **Lista de pendências** (endereço – código parcialmente montado)
  - Ponteiro para próxima pendência
- **Tabela de mnemônicos e códigos**
  - Mnemônico
  - Código
  - Classe
- **Tabela de equivalências**
  - Símbolo
  - Link
- **Área de saída**
  - Bloco de código objeto gerado

Pseudocódigo para montador de um passo

- **Iniciar as áreas de dados do montador: CI=0; NLINHA=1;**
- **Repetir as operações seguintes ate o final da montagem do programa:**
  - **Ler uma linha do texto simbólico**
  - **Se comentário: listar, e voltar a ler outra linha até que não seja comentário.**
  - **Se houver rótulo no campo de rótulos**
    - **tratar o rótulo como no montador de dois passos**
  - **Se houver elementos na lista de pendências associada,**
    - **gerar, para cada pendência, o código correspondente ("*backtrack*")**
  - **Analisar o campo do mnemônico como no montador de dois passos**
  - **Se houver referência a símbolos indefinidos,**
    - **criar a trinca (CI, OP, D), com OP= código de operação da instrução e D= deslocamento imposto pelo operando**
    - **inseri-la na lista de pendências do símbolo referenciado**
  - **Se for pseudo-instrução, tratá-la como no montador de dois passos**
  - **Emitir mensagem de erro se CI ultrapassar os limites da memorla**
  - **Imprimir a linha simbólica e o código gerado (se possível)**
  - **Incrementar NLINHA**
  - **Imprimir a tabela de símbolos e informações complementares sobre o código-objeto**

Áreas de Dados usadas pelo montador



Exemplo da estrutura do bloco de saída

- O trecho abaixo representa um fragmento de um programa, com a finalidade de ilustrar algumas situações típicas que costumam ser encontradas nos programas em notação simbólica absoluta, e que devem ser tratadas pelo montador.

Endereço	Código	Rótulo	Mnemônico	Operando	Comentários
			@	/0100	
0100	00	DADO	K	0	
0101	00	AUX	K	0	
0102	81 01	INIC	LD	AUX	; primeira instrução executável do programa
⋮	⋮		⋮		
0124	91 01		MM	AUX	; salva acumulador em AUX
0126	81 00	LOOP	LD	DADO	; esta instrução copia DADO para o acumulador
⋮	⋮		⋮		
01AF	01 26		JP	LOOP	; esta instrução fecha o loop
0102			#	INIC	; indica INIC como endereço de partida do prog.

Observações sobre o bloco de saída

- Como já deve ter sido notado, é **incremental** a **geração do código objeto** à imagem da memória, alguns bytes para cada linha do programa fonte.
- Para que o programa objeto seja **compatível com o loader**, seu formato não é à imagem da memória, mas, blocado em partes referentes a áreas relativamente pequenas de código, e incluindo um byte de checksum, para reduzir a probabilidade de leitura incorreta dos dados.
- Assim sendo, e levando em conta que blocos muito pequenos acarretam o aumento desnecessário do tamanho do código objeto, opta-se por **acumular na memória do montador** uma série de códigos vizinhos à imagem da memória até que estejam acumulados bytes suficientes para formar um bloco de tamanho aceitável. Nessa ocasião, o bloco é **transferido para o arquivo de saída**, e um novo bloco, vazio, é criado e passa a ser preenchido pelo montador, a partir de uma nova origem.
- Durante a montagem, o aparecimento de instruções ou de pseudo instruções que **modificam a origem do código gerado** devem, portanto, **iniciar um novo bloco a partir da nova origem**, sendo portanto necessário **forçar o encerramento de um eventual bloco** parcialmente formado na memória do montador, para que o código já gerado, nele contido, não seja sobrescrito, e portanto perdido.

LÓGICA GERAL DE MONTADORES (DE UM OU DOIS PASSOS)

Lógica geral do montador (de 1 ou 2 passos)

- Fazer Contador de Instruções (C.I.) igual a 0
- Fazer Passo igual a 1
- Ler uma linha do programa-fonte
- Se for linha de comentário:
  - Se Passo for igual a 1, ignorar a linha lida
  - Se Passo for igual a 2, então listar a linha

Se a linha tiver rótulo

- Procurar o rótulo da linha na Tabela de Símbolos
- Se já **existe** na tabela, e está **definido**,
  - reportar **erro de múltipla definição** de rótulo
- Se já **existe** na tabela, mas ainda está **indefinido**,
  - **defini-lo** c/ endereço apontado pelo **contador de instruções**
- Se ainda **não existe**
  - **inserir na tabela** e **defini-lo** c/ o endereço apontado pelo **contador de instruções**
- Marcar na Tabela de Símbolos como rótulo definido
- Incluir o **número da linha** associando-a à **definição** do rótulo na tabela de **referências cruzadas**

Analisar o campo de mnemônico

- Procurar na tabela de mnemônicos
  - Se não for achado, **reportar erro: mnemônico inválido**
- Atualizar o contador de instruções, conforme o tipo de mnemônico encontrado
- Se o mnemônico exigir operando:
  - Analisar o campo de operando:
    - **Incluir** eventuais símbolos novos **na tabela de símbolos**
    - Os que não constarem na tabela, **marcar como indefinidos**
    - **Atualizar** a tabela de **referências cruzadas** (referências)
  - **Avaliar eventual expressão** encontrada no operando
  - Se for passo 2 e houver código-objeto associado a gerar
    - **Montar código objeto** correspondente

Tratamento de Pseudo Instruções

- Em **montadores absolutos**, costumam ser encontradas as **pseudo-instruções** seguintes:
  - **ORG** – (nova origem) modifica o contador de instruções conforme o valor do operando **(@)**
  - **BLOC** – (reserva de área) modifica contador de instruções para contabilizar a área reservada **($)**
  - **DB, DW, DA** – (preenche memória com constante) se passo igual 2, gerar código objeto **(K)**
  - **EQU** – (define sinônimos) se passo igual a 1, atualizar a tabela de equivalências
  - **END** – (indica final físico do programa fonte) se passo for igual a 1, fazer passo igual a 2. Se não, encerrar os trabalhos do montador **(#)**
- Voltar à leitura de nova linha.

ORG (origin)

- Determina nova origem para o código a ser gerado em seguida pelo montador:
  - Em montadores absolutos, o operando deve ser obrigatoriamente absoluto.
  - Em montadores relocáveis, pode ser relocável, absoluto, simbólico, relativo
  - Tratamento:
    Se passo for igual a 2 e houver no bloco de saída do código objeto algum código ainda não gerado em meio externo, gerar o bloco devidamente, e esvaziá-lo;
    Modificar o contador de instruções do montador, de acordo com o valor do operando que estiver sendo especificado.

BLOC (define memory block)

- Esta pseudo-instrução determina a reserva de uma área de memória de comprimento estabelecido, sem preenchimento de dados, disponibilizando-a para uso pelo programa objeto.
- O operando deve ter um valor numérico inteiro não negativo, pois refere-se ao número de palavras de memória a ser reservado.
- Tratamento:
  Equivalente à definição de nova origem no endereço obtido adicionando-se ao contador de instruções o tamanho da área que estiver sendo reservada.
  Em ambos os passos da montagem, atualizar o contador de instruções, adicionando-lhe o valor declarado no seu operando.

DB (define byte)

- Esta pseudo instrução destina-se a preencher o endereço de memória apontado pelo contador de instruções corrente e o seguinte, com o valor (um byte) associado ao seu operando.
- O valor do operando dessa pseudo instrução deve ser numérico, e expresso como número binário de um byte (oito bits).
- Tratamento: se passo for igual a 2,
  - Gerar código-objeto preenchendo um byte, com o valor do operando, no endereço de memória apontado pelo contador de instruções.
  - Atualizar o contador de instruções, incrementando-o de uma unidade.

DW (define word)

- Esta pseudo instrução destina-se a preencher o endereço de memória apontado pelo contador de instruções corrente e o seguinte, com o valor (dois bytes) associado ao seu operando.
- O valor do operando dessa pseudo instrução deve ser numérico, e expresso como número binário de dois bytes.
- Tratamento: se passo for igual a 2,
  - Gerar código-objeto preenchendo dois bytes, com o valor do operando, nos endereços de memória apontado pelo contador de instruções e seguinte.
  - Atualizar o contador de instruções incrementando-o de duas unidades.

DA (define address)

- Esta pseudo-instrução destina-se a preencher o endereço de memória apontado pelo contador de instruções corrente e o seguinte, com a representação binária de um endereço (dois bytes) associado ao seu operando, a ser usado como ponteiro pelo programa.
- O valor do operando dessa pseudo-instrução deve ser um endereço absoluto, e expresso como um número binário de dois bytes.
- Tratamento: se passo for igual a 2,
  - Gerar código-objeto preenchendo dois bytes, com o valor do operando, nos endereços de memória apontado pelo contador de instruções e seguinte.
  - Atualizar o contador de instruções incrementando-o de duas unidades.

EQU (define equivalence)

- Esta pseudo-instrução permite determinar a equivalência (sinônimos) entre novos nomes e endereços associados a um ou mais símbolos definidos no programa-fonte.
- Seu operando precisa ser obrigatoriamente uma expressão simbólica que represente algum endereço de memória, de qualquer tipo.
- Tratamento: se passo for igual a 1, atualizar a tabela de equivalências

END (end mark for source program)

- Esta pseudo-instrução permite ao programador informar ao montador que foi atingido o final físico do programa-fonte.
- Seu operando deve ser um rótulo definido no programa, e deve referir-se a um endereço de memória, relativo a um rótulo do programa, que fornece a informação do endereço a partir do qual está previsto o início da execução do programa.
- Tratamento da pseudo-instrução FIM
  - Se passo for igual a 1, fazer passo igual a 2.
  - Se não, encerrar a execução do montador.