

PCS 3216 – Sistemas de Programação

14 - Programas relocáveis: Preparação para a execução

Um exemplo ilustrativo do processo

Construção de programas relocáveis

- Como foi estudado, um programa pode ser construído de muitas maneiras para ser executado no computador.
- Uma das formas mais naturais de fazer isso consiste em particionar o programa em **módulos relocáveis**, como é feito no caso da decomposição funcional:
 - Um **programa principal** relocável
 - Um **conjunto de subrotinas** relocáveis
- Com tal procedimento, as diversas partes do programa podem ser **desenvolvidas independentemente**, por autores ou equipes diferentes, em épocas diferentes, e até em linguagens de programação diferentes.

EXEMPLO ILUSTRATIVO

Um exemplo

- O módulo relocável que compõe o **programa principal** (*main program*) constitui a **raiz de uma árvore** de módulos relocáveis interdependentes
- Seu código-objeto relocável caracteriza-se por conter um metadado que informa ao sistema operacional o endereço dessa primeira instrução executável do programa
- Cada módulo deve conter metadados, na forma de rótulos que identificam endereços simbólicos que representam pontos de acesso (*entry points, exports*) e referências externas (*externals, imports*)
- Neste exemplo, será mostrada a obtenção de uma versão relocável de um programa absoluto já apresentado na disciplina, o qual efetua o cálculo do quadrado de um número inteiro n efetuando a somatória de uma série de números ímpares sucessivos: $n^2 = 1+3+5+\dots+(2n-1)$

O programa absoluto original

	ORG	/010
INIC	LD	UM
	MM	CONT
	MM	IMPAR
	MM	N2
LOOP	LD	CONT
	SUB	N
	JZ	FORA
	LD	CONT
	ADD	UM
	MM	CONT
	LD	IMPAR
	ADD	DOIS
	MM	IMPAR
	ADD	N2
	MM	N2
	JP	LOOP
FORA	HM	FORA
UM	K	/01
DOIS	K	/02
IMPAR	K	/00
N	K	/04
N2	K	/00
CONT	K	/00
	FIM	INIC

- O programa ao lado implementa um algoritmo que efetua o cálculo de $N2 = N^2$ a partir de N, pela soma da série de N ímpares sucessivos $N^2 = 1 + 3 + \dots + (2.N - 1)$
- FORA é um *trap* onde o programa pára após sua execução.
- INIC é o endereço inicial de partida do programa.
- UM=1 e DOIS=2 são constantes.
- IMPAR implementa cada um dos ímpares a somar.
- CONT é o contador de iterações do algoritmo.

Código do programa absoluto

1	/010			ORG	/010
2	/010	8032	INIC	LD	UM
3	/012	9037		MM	CONT
4	/014	9034		MM	IMPAR
5	/016	9036		MM	N2
6	/018	8037	LOOP	LD	CONT
7	/01A	5035		SUB	N
8	/01C	1030		JZ	FORA
9	/01E	8037		LD	CONT
10	/020	4032		ADD	UM
11	/022	9037		MM	CONT
12	/024	8034		LD	IMPAR
13	/026	4033		ADD	DOIS
14	/028	9034		MM	IMPAR
15	/02A	4036		ADD	N2
16	/02C	9036		MM	N2
17	/02E	0018		JP	LOOP
18	/030	C030	FORA	HM	FORA
19	/032	01	UM	K	/01
20	/033	02	DOIS	K	/02
21	/034	00	IMPAR	K	/00
22	/035	04	N	K	/04
23	/036	00	N2	K	/00
24	/037	00	CONT	K	/00
25	/010			FIM	INIC

DECOMPOSIÇÃO EM MÓDULOS

Divisão do programa em módulos

	ORG	/010
INIC	LD	UM
	MM	CONT
	MM	IMPAR
	MM	N2
LOOP	LD	CONT
	SUB	N
	JZ	FORA
	LD	CONT
	ADD	UM
	MM	CONT
	LD	IMPAR
	ADD	DOIS
	MM	IMPAR
	ADD	N2
	MM	N2
	JP	LOOP
FORA	HM	FORA
UM	K	/01
DOIS	K	/02
IMPAR	K	/00
N	K	/04
N2	K	/00
CONT	K	/00
	FIM	INIC

Principal

– Consolida os módulos

Início

– Impõe valores iniciais

Repetição

– Controla o número de execuções do núcleo

Núcleo

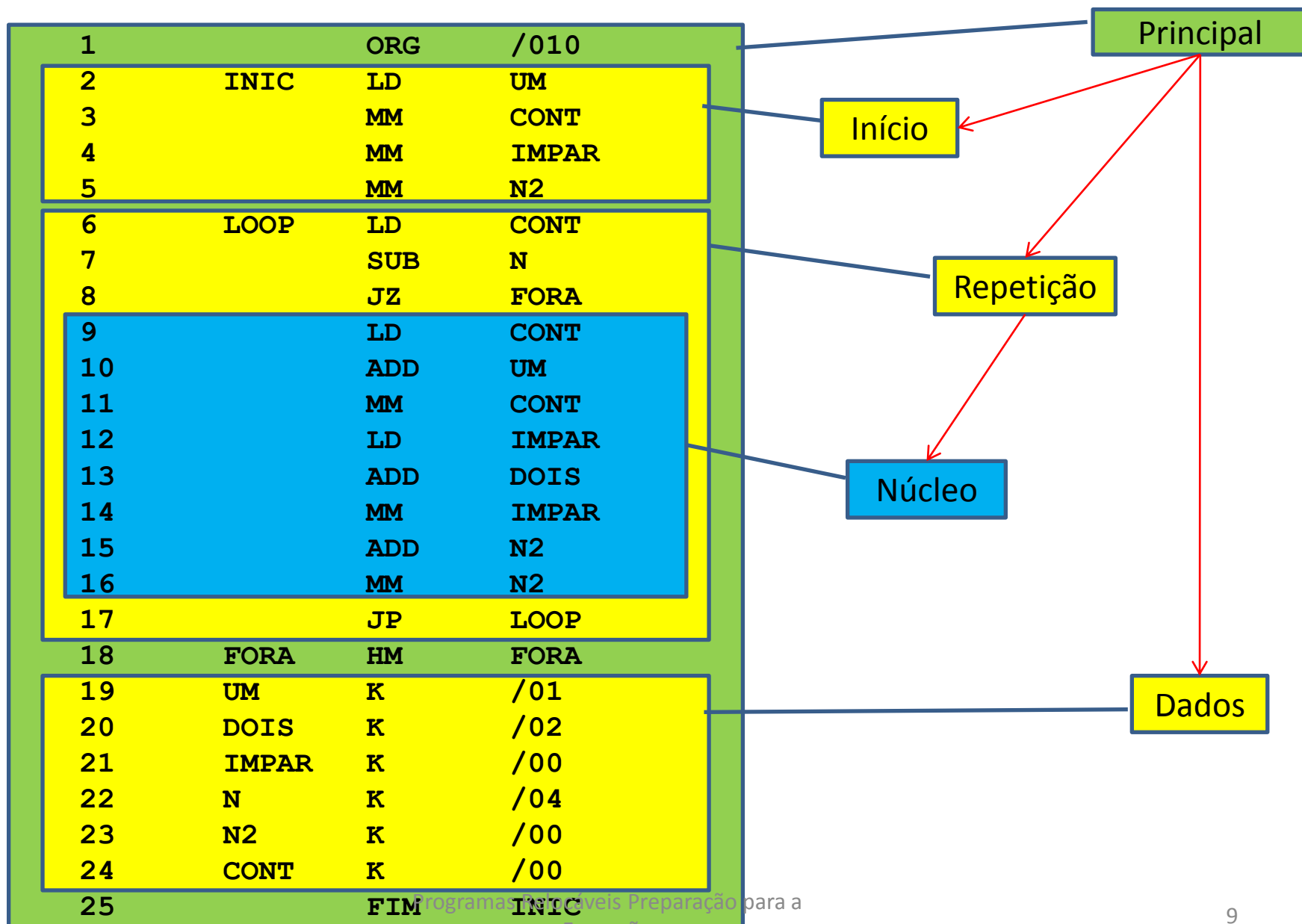
– Efetua uma só iteração

Dados

– Variáveis de trabalho.

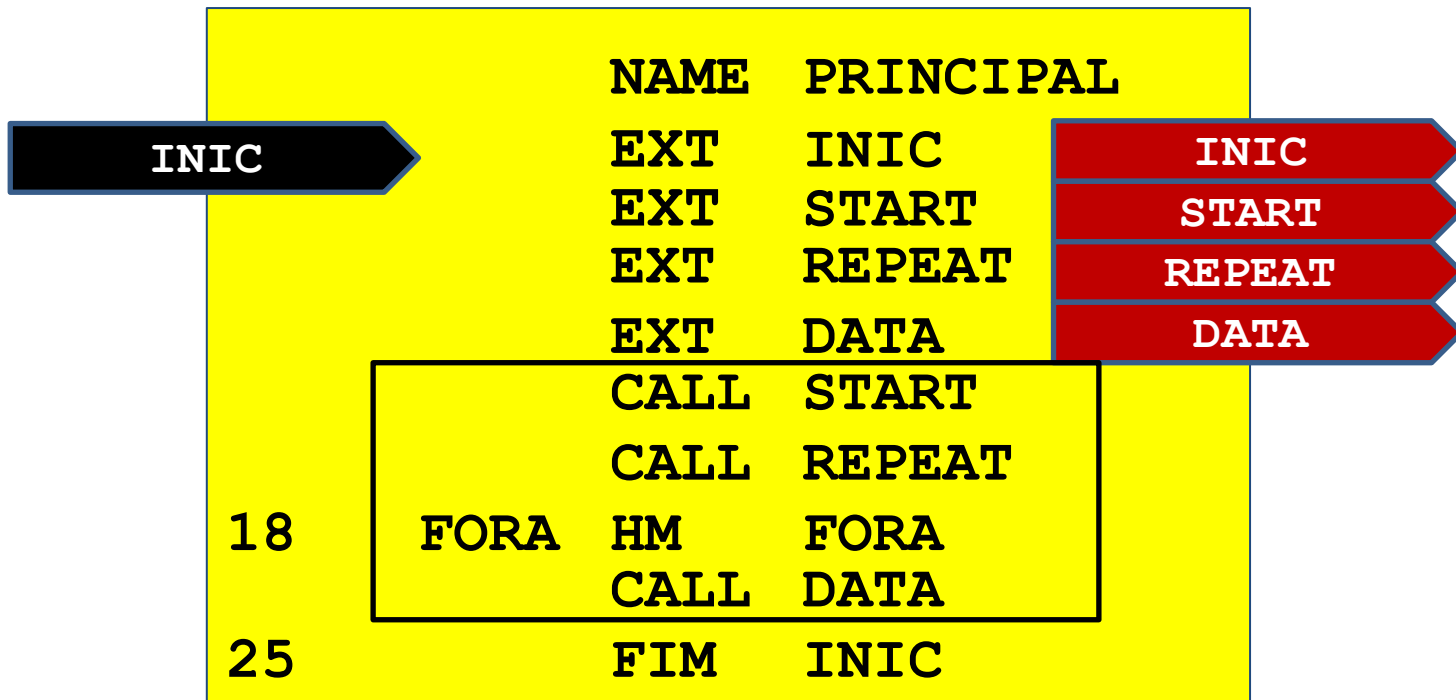
Entrada: N; Saída: $N2 = N^2$.

Símbolos especiais



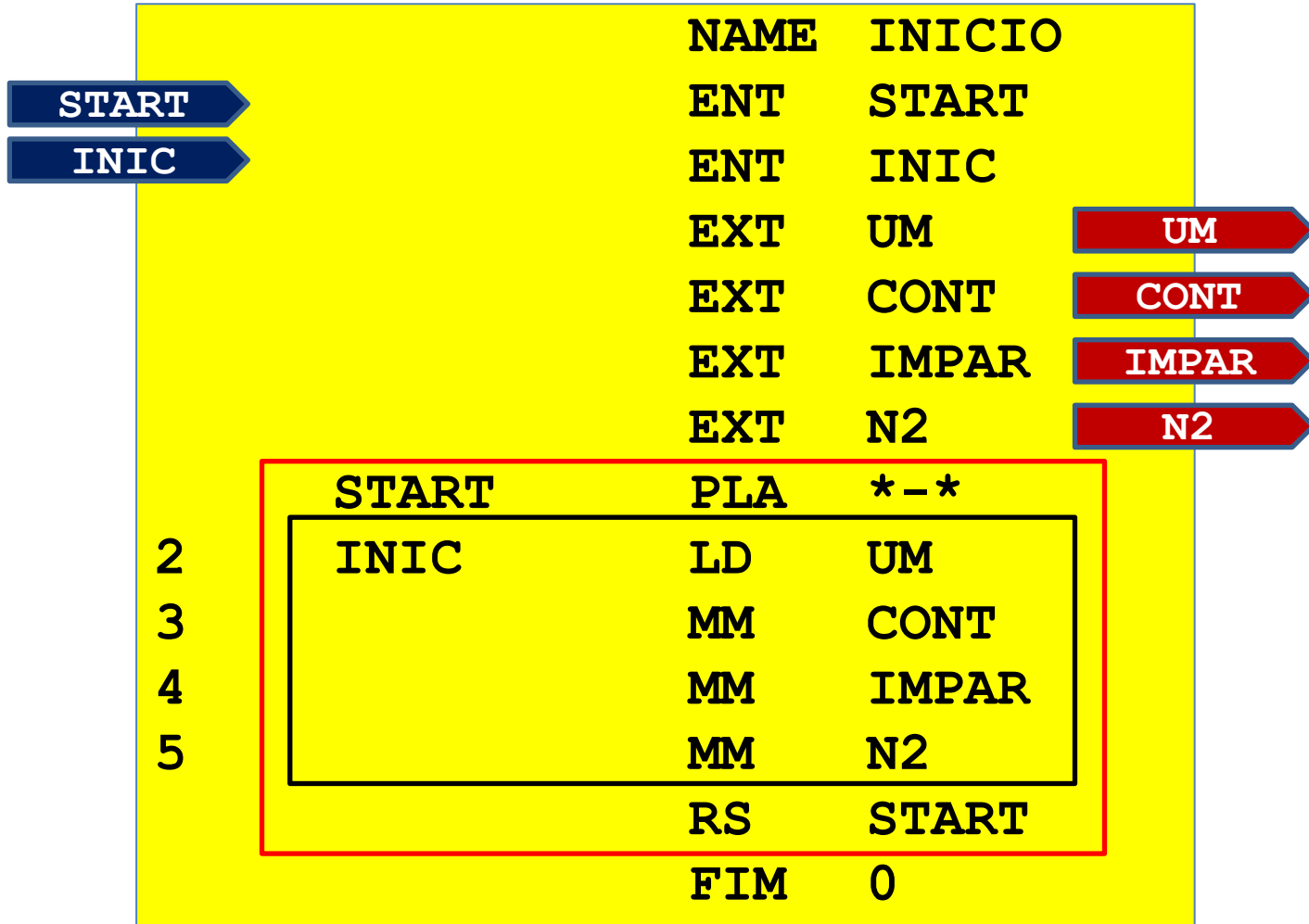
OS MÓDULOS

Principal





Início



REPEAT

REPETICAO

N

CONT

KERNEL

Repetição

REPEAT

NAME REPETICAO

ENT REPEAT

EXT KERNEL

EXT CONT

EXT N

KERNEL

CONT

N

	REPEAT	PLA	* - *
6	LOOP	LD	CONT
7		SUB	N
8		JZ	RETORNO
		CALL	KERNEL
17		JP	LOOP
	RETORNO	RS	REPEAT
25		FIM	0

KERNEL

NUCLEO

DOIS

CONT

N2

IMPAR

UM

Núcleo

KERNEL

NAME	NUCLEO
ENT	KERNEL
EXT	CONT
EXT	UM
EXT	DOIS
EXT	IMPAR
EXT	N2

CONT

UM

DOIS

IMPAR

N2

KERNEL

PLA * - *

9

LD CONT

10

ADD UM

11

MM CONT

12

LD IMPAR

13

ADD DOIS

14

MM IMPAR

15

ADD N2

16

MM N2

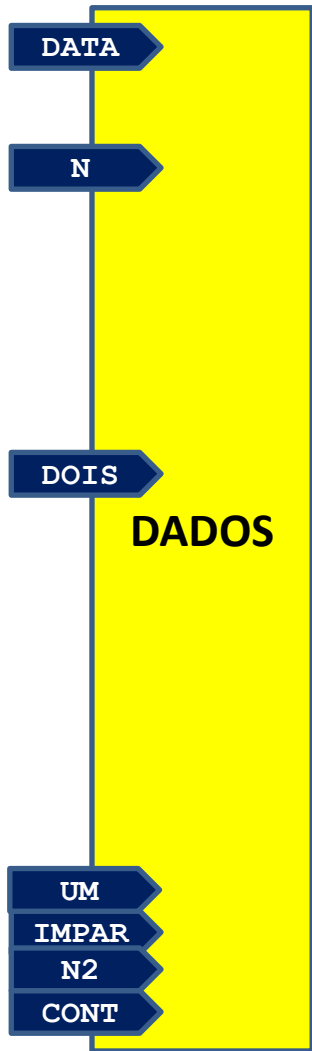
RS KERNEL

Programas Relocáveis Preparação para a Execução

FIM

0

Dados



	NAME	DADOS
DATA	ENT	DATA
UM	ENT	UM
DOIS	ENT	DOIS
IMPAR	ENT	IMPAR
N	ENT	N
N2	ENT	N2
CONT	ENT	CONT

	DATA	PLA	* - *
		RS	DATA
19	UM	K	/01
20	DOIS	K	/02
21	IMPAR	K	/00
22	N	K	/04
23	N2	K	/00
24	CONT	K	/00
	END		0

COLETA DAS REFERÊNCIAS SIMBÓLICAS

Referências simbólicas entre os módulos

PRINCIPAL

Endereço de execução do programa: Inic

Externals: Inic, Start, Repeat, Data

INÍCIO

Entry: Start, Inic

Externals: Um, Cont, Impar, N2

REPETIÇÃO

Entry: Repeat

Externals: N, Cont, Kernel

DADOS

Entry: Data, Um, Dois, Impar, N, N2, Cont

Externals: ---

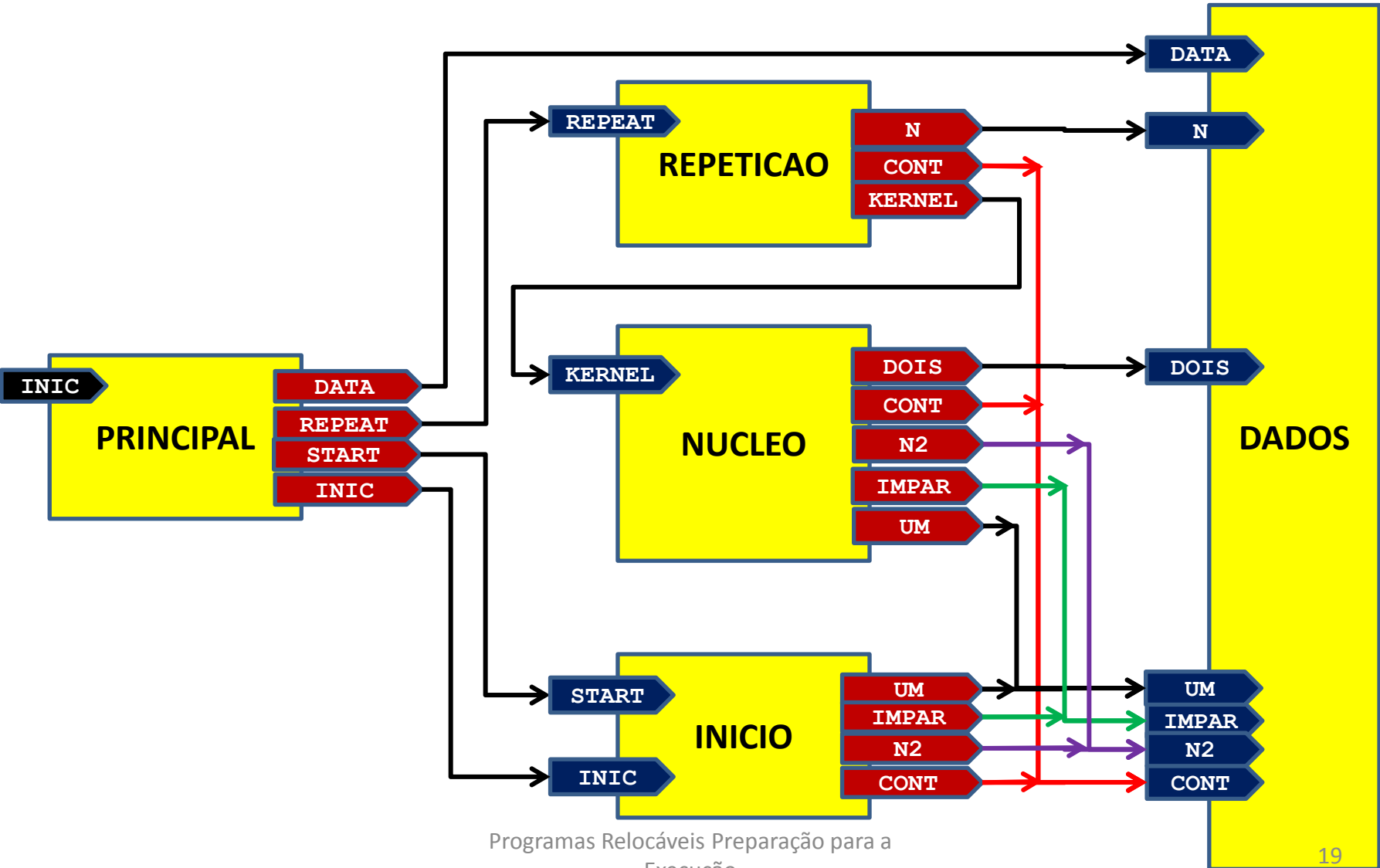
NÚCLEO

Entry: Kernel

Externals: Cont, Um, Impar, Dois, N2

DIAGRAMA DE CONEXÕES

Referências simbólicas entre módulos



PREPARAÇÃO DO PROGRAMA PARA QUE POSSA SER EXECUTADO

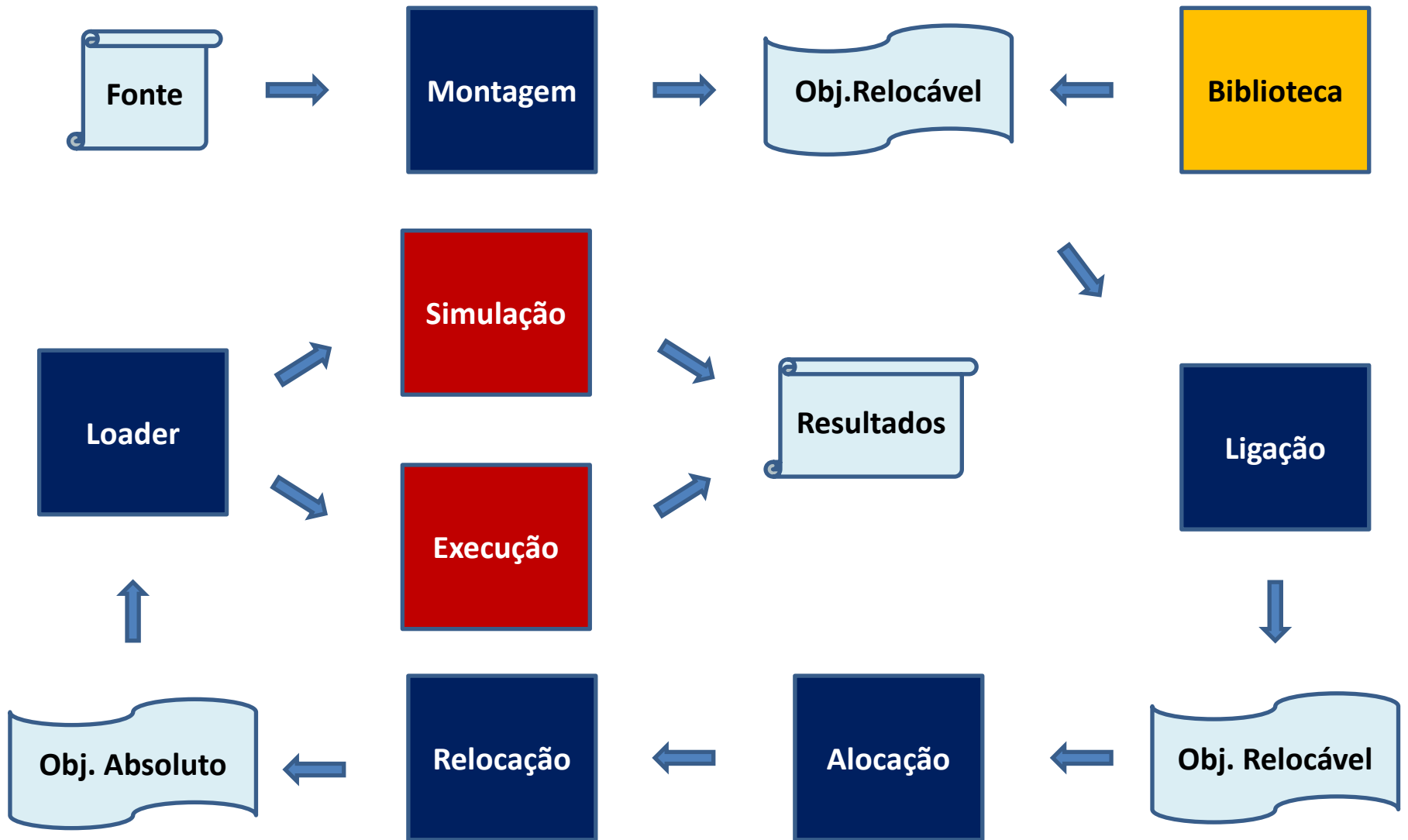
Programas-fonte relocáveis

- Da decomposição que acabamos de fazer, resultaram **cinco módulos**, na forma de programas-fonte **relocáveis**, um para o **programa principal** e os demais para as quatro **subrotinas** do programa.
- A primeira etapa para a obtenção de um programa executável é a conversão desses textos-fonte nos correspondentes códigos-objeto relocáveis.
- Essa fase do processamento é executada pelo **montador relocável**, que converte cada um dos cinco módulos-fonte nos códigos-objeto relocáveis equivalentes, nos quais
 - os **mnemônicos simbólicos** são traduzidos para a forma de códigos relocáveis, em geral binários ou hexadecimais.
 - As **pseudos** são traduzidas para a forma dos metadados correspondentes no código-objeto relocável produzido.

Preparação de um programa relocável

- O processo de preparação de um programa relocável para ser processado no computador envolve as etapas:
 - Construção do **programa-fonte** pelo programador
 - **Montagem** de cada componente relocável do programa
 - Coleta dos programas-objeto relocáveis em **bibliotecas**
 - **Ligação** do conjunto de códigos relocáveis
 - **Alocação** de área de memória para o programa
 - **Relocação** do programa consolidado pelo ligador
 - Obtenção do **programa-objeto absoluto** final, executável
 - **Simulação** ou **execução** do programa no computador.
- O próximo slide ilustra este processo, e os seguintes detalham cada uma dessas etapas.

Processo de preparação para a execução



ETAPAS DA PREPARAÇÃO DO PROGRAMA PARA SEU PROCESSAMENTO

Montagem

- Os cinco programas-fonte relocáveis assim obtidos devem ser inicialmente submetidos ao **montador relocável**, que os converte para a forma de códigos-objeto relocáveis: um programa principal e quatro subrotinas.
- Ao contrário de um código-objeto absoluto, esta versão relocável ainda não pode ser executada, dado que nela estão presentes referências relativas e simbólicas a serem resolvidas.
- O montador se encarrega de eliminar mnemônicos e referências simbólicas internas aos programas-fonte, convertendo-os para a forma de códigos-objeto relocáveis. Isso é feito separadamente para cada um dos cinco textos-fonte.

Bibliotecas

- Programas-objeto relocáveis produzidos por montadores e por outros componentes do sistema de programação (tais como ligadores, compiladores e outros) são, muitas vezes, organizados em bibliotecas públicas ou particulares, para facilitar o seu manuseio para o usuário.

Ligação

- Os programas-objeto relocáveis geralmente contêm referências simbólicas a outros, para que seja facilitada a sua reutilização como elemento de uma biblioteca pública, ou para efeito de intercâmbio entre programadores.
- Tais referências simbólicas entre módulos devem ser resolvidas pelo **ligador**, o qual deve eliminá-las, transformando-as em referências relativas.
- Dado um programa principal, a meta do ligador é a de eliminar todas as referências simbólicas a outros módulos relocáveis, e obter um único módulo relocável que os consolide na forma de um código relocável **sem referências simbólicas**, e apenas com referências relativas a uma **base única de relocação**.

Alocação

- O produto da ligação é um programa-objeto relocável sem referências simbólicas, e portanto completo, do ponto de vista dos seus módulos componentes.
- Sabe-se portanto exatamente a quantidade de memória de que necessita para que possa ser executado.
- Na operação chamada **alocação**, esta informação é passada ao **sistema operacional**, que escolhe os endereços mais adequados à execução do programa.
- O programa ainda não pode ser carregado na memória, dado que nele persistem referências relativas não resolvidas, exigindo uma operação de relocação que as converta em referências absolutas.

Relocação

- Neste ponto, tendo sido eliminadas do programa todas as referências simbólicas, as referências relativas restantes devem ser convertidas em referências absolutas transformando o programa relocável em um equivalente absoluto, pronto para a execução.
- Esta operação, denominada relocação, é feita pelo **relocador**, que justapõe os códigos em um único, uniformizando as referências relativas ao início de cada módulo em referências relativas à base única de relocação, e convertendo-as em referências absolutas pela adição do endereço determinado pelo sistema operacional na operação de alocação.
- O resultado dessa etapa é um programa-objeto absoluto, executável a partir do endereço declarado pelo programador na pseudo-instrução **FIM** do seu programa principal.

Carregamento na memória

- O código-objeto absoluto assim obtido, em tudo similar ao construído por um montador absoluto, pode agora ser carregado na memória pelo **loader** na memória de uma máquina (real ou virtual), para que o programa possa ser finalmente processado (executado ou simulado, respectivamente).
- Esta operação é comandada pelo **sistema operacional**, o qual, neste ponto, já possui as informações dos endereços de alocação e de execução do programa.

Execução

- Uma vez carregado o programa absoluto, pelo loader, na memória do processador, o sistema operacional já pode inseri-lo em sua escala de execução, da forma que for mais conveniente.
- Ao término da execução, o programa sai do sistema, e o sistema operacional reassume o controle do processador.

O PROCESSO DE LIGAÇÃO NO EXEMPLO

- Vamos acompanhar o processo de ligação dos módulos do nosso exemplo.
- Inicialmente, o **ligador** deve receber o **programa principal**, e dele extrair os metadados, que são guardados na tabela de externals do ligador.
- Ao início do processamento do ligador, essa tabela se encontra vazia, assim como o programa ligado que se deseja construir.

Tab. de externals, vazia ao início da ligação:

Símbolo	Endereço relativo	Resolvido?
---------	-------------------	------------

Programa ligado, inicialmente vazio

Deslocamento para o módulo Principal = 0

Comprimento total do programa ligado até esta ocasião = 0

Programas Relocáveis: Preparação para
Execução

Início do processo de ligação

- Para dar partida ao processo de ligação, é necessário iniciar apresentando ao ligador o módulo relocável do programa principal.
- Devem ser extraídas as seguintes informações:
 - O endereço de execução do programa, a ser informado ao sistema operacional.
 - Código-objeto associado ao módulo, para ser anexado ao programa ligado final desejado.
 - Os metadados referentes a entry-points e externals, para serem guardados na tabela de externals do ligador.
 - O comprimento da área de código do módulo, para o cálculo do valor da base de relocação do módulo.

INIC

PRINCIPAL

DATA

REPEAT

START

INIC

Principal

INIC

NAME PRINCIPAL

EXT INIC

EXT START

EXT REPEAT

EXT DATA

CALL START

CALL REPEAT

18

FORA HM FORA

CALL DATA

25

FIM INIC

INIC

START

REPEAT

DATA

Comprimento deste código = /008 bytes

```

000R      CALL  START
002R      CALL  REPEAT
004R FORA  HM   FORA
006R      CALL  DATA

```

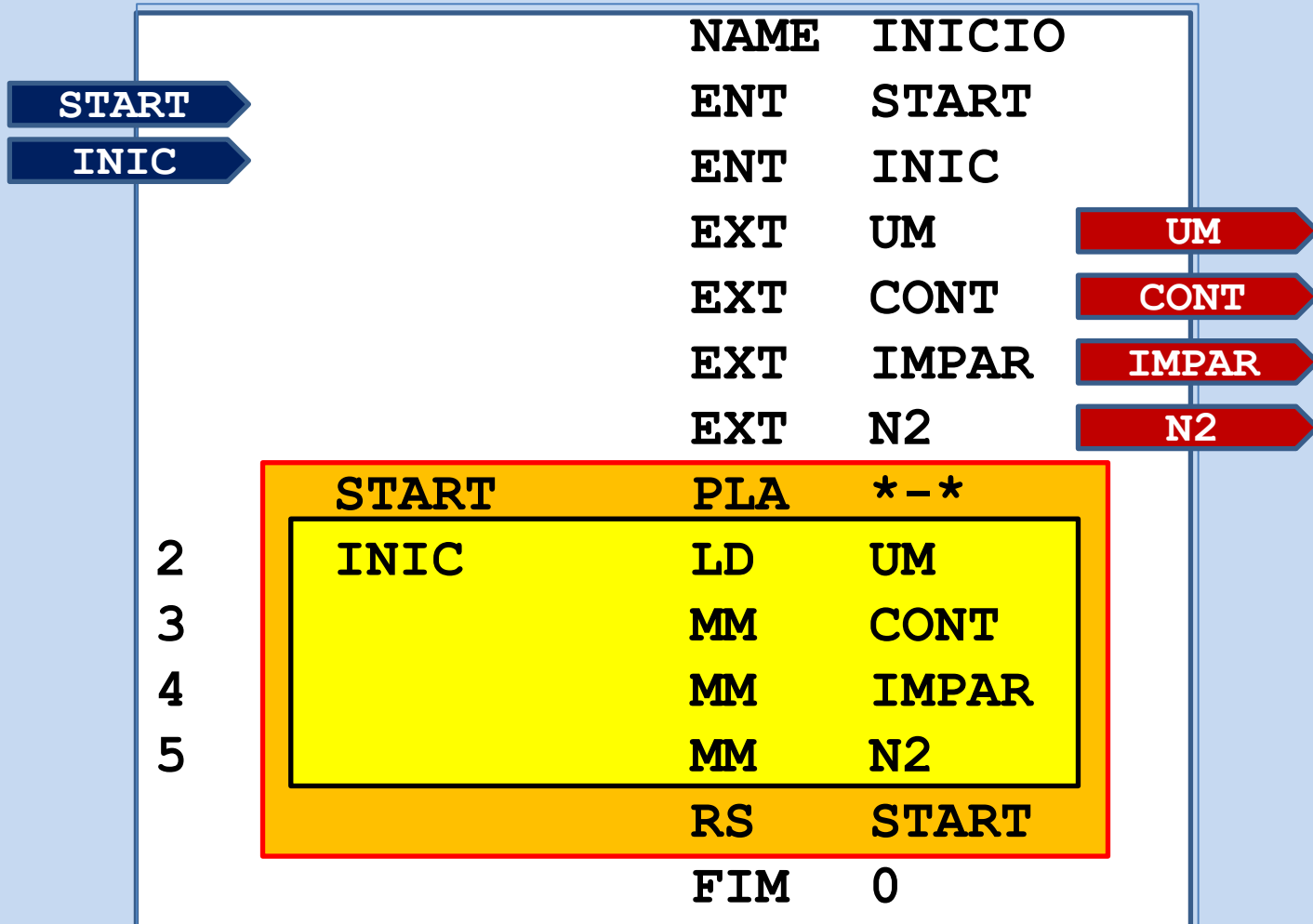
Programa ligado parcial, após Principal

```
0000          CALL  START
0002          CALL  REPEAT
0004  FORA    HM    FORA
0006          CALL  DATA
```

Tab. de External partial, após Principal

Símbolo	Endereço relativo	Resolvido?
INIC		não
START		não
REPEAT		não
DATA		não

Início



Programa ligado parcial, após Início

0000		CALL	START
0002		CALL	REPEAT
0004	FORA	HM	FORA
0006		CALL	DATA
0008	START	PLA	* - *
000A	INIC	LD	UM
000C		MM	CONT
000E		MM	IMPAR
0010		MM	N2
0012		RS	START

Tabela de External partial, após Início

Símbolo	Endereço relativo	Resolvido?
INIC	009	sim
START	007	sim
REPEAT		não
DATA		não
UM		não
IMPAR		não
N2		não
CONT		não

REPEAT

REPETICAO

N

CONT

KERNEL

Repetição

REPEAT

NAME REPETICAO

ENT REPEAT

EXT KERNEL

EXT CONT

EXT N

KERNEL

CONT

N

6

7

8

17

25

REPEAT

LOOP

RETORNO

PLA

LD

SUB

JZ

CALL

JP

RS

FIM

* - *

CONT

N

RETORNO

KERNEL

LOOP

REPEAT

0

Programa ligado parcial, após Repetição

0000		CALL	START
0002		CALL	REPEAT
0004	FORA	HM	FORA
0006		CALL	DATA
0008	START	PLA	* - *
000A	INIC	LD	UM
000C		MM	CONT
000E		MM	IMPAR
0010		MM	N2
0012		RS	START
0014	REPEAT	PLA	* - *
0016	LOOP	LD	CONT
0018		SUB	N
001A		JZ	RETORNO
001C		CALL	KERNEL
001E		JP	LOOP
0020	RETORNO	RS	REPEAT

Tabela de External partial, após Repetição

Símbolo	Endereço relativo	Resolvido?
INIC	009	sim
START	007	sim
REPEAT	013	sim
DATA		não
UM		não
IMPAR		não
N2		não
CONT		não
N		não
KERNEL		não

KERNEL

NUCLEO

DOIS

CONT

N2

IMPAR

UM

Núcleo

KERNEL

NAME	NUCLEO
ENT	KERNEL
EXT	CONT
EXT	UM
EXT	DOIS
EXT	IMPAR
EXT	N2

CONT

UM

DOIS

IMPAR

N2

KERNEL

PLA

* - *

9

LD

CONT

10

ADD

UM

11

MM

CONT

12

LD

IMPAR

13

ADD

DOIS

14

MM

IMPAR

15

ADD

N2

16

MM

N2

RS

KERNEL

Programas Relocáveis Preparação para a
Execução

FIM

0

Programa ligado parcial, após Núcleo

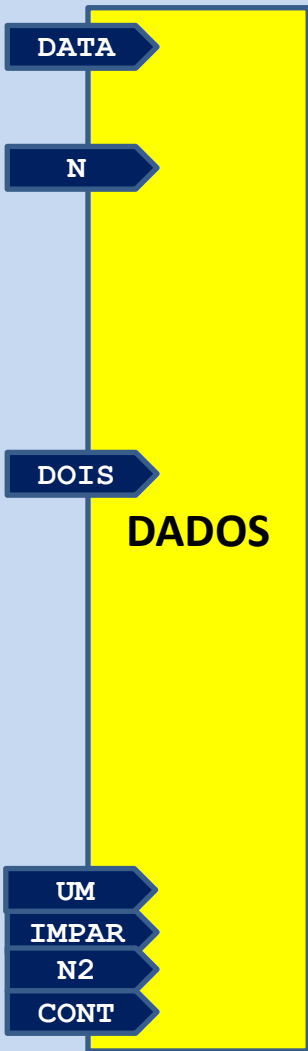
0000		CALL	START
0002		CALL	REPEAT
0004	FORA	HM	FORA
0006		CALL	DATA
0008	START	PLA	*-*
000A	INIC	LD	UM
000C		MM	CONT
000E		MM	IMPAR
0010		MM	N2
0012		RS	START
0014	REPEAT	PLA	*-*
0016	LOOP	LD	CONT
0018		SUB	N
001A		JZ	RETORNO
001C		CALL	KERNEL
001E		JP	LOOP
0020	RETORNO	RS	REPEAT

0022	KERNEL	PLA	*-*
0024		LD	CONT
0026		ADD	UM
0028		MM	CONT
002A		LD	IMPAR
002C		ADD	DOIS
002E		MM	IMPAR
0030		ADD	N2
0032		MM	N2
0034		RS	KERNEL

Tabela de External partial, após Núcleo

Símbolo	Endereço relativo	Resolvido?
INIC	009	sim
START	007	sim
REPEAT	013	sim
DATA		não
UM		não
IMPAR		não
N2		não
CONT		não
N		não
KERNEL	021	sim

Dados



	NAME	DADOS
DATA	ENT	DATA
UM	ENT	UM
DOIS	ENT	DOIS
IMPAR	ENT	IMPAR
N	ENT	N
N2	ENT	N2
CONT	ENT	CONT

	DATA	PLA	* - *
		RS	DATA
19	UM	K	/01
20	DOIS	K	/02
21	IMPAR	K	/00
22	N	K	/04
23	N2	K	/00
24	CONT	K	/00

END 0

Programa ligado completo, após Dados

0000		CALL	START
0002		CALL	REPEAT
0004	FORA	HM	FORA
0006		CALL	DATA
0008	START	PLA	*-*
000A	INIC	LD	UM
000C		MM	CONT
000E		MM	IMPAR
0010		MM	N2
0012		RS	START
0014	REPEAT	PLA	*-*
0016	LOOP	LD	CONT
0018		SUB	N
001A		JZ	RETORNO
001C		CALL	KERNEL
001E		JP	LOOP
0020	RETORNO	RS	REPEAT

0022	KERNEL	PLA	*-*
0024		LD	CONT
0026		ADD	UM
0028		MM	CONT
002A		LD	IMPAR
002C		ADD	DOIS
002E		MM	IMPAR
0030		ADD	N2
0032		MM	N2
0034		RS	KERNEL
0036	DATA	PLA	*-*
0038		RS	DATA
003A	UM	K	/01
003B	DOIS	K	/02
003C	IMPAR	K	/00
003D	N	K	/04
003E	N2	K	/00
003F	CONT	K	/00

Tabela de Externalis completa, após Dados

Símbolo	Endereço relativo	Resolvido?
INIC	009	sim
START	007	sim
REPEAT	013	sim
DATA	035	sim
UM	039	sim
IMPAR	03B	sim
N2	03D	sim
CONT	03E	sim
N	03C	sim
KERNEL	021	sim
DOIS	03A	sim