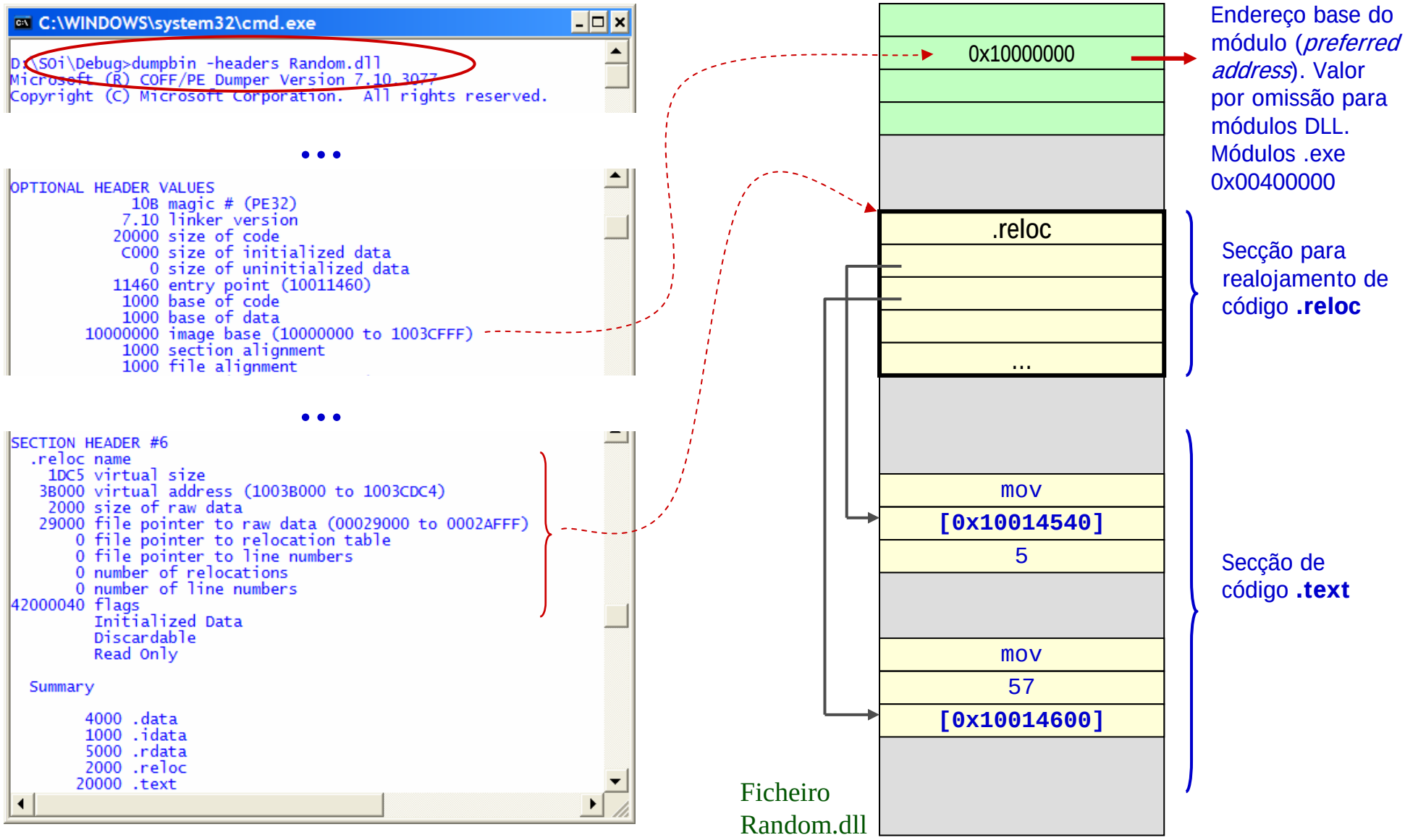
Dynamic Link Library
(DLL)
Rebase e *Bind*

- *Jeffrey Richter, Christophe Nasarre,* ***Windows via C/C++****, Fifth Edition, Microsoft Press, 2008 [cap. 20]*

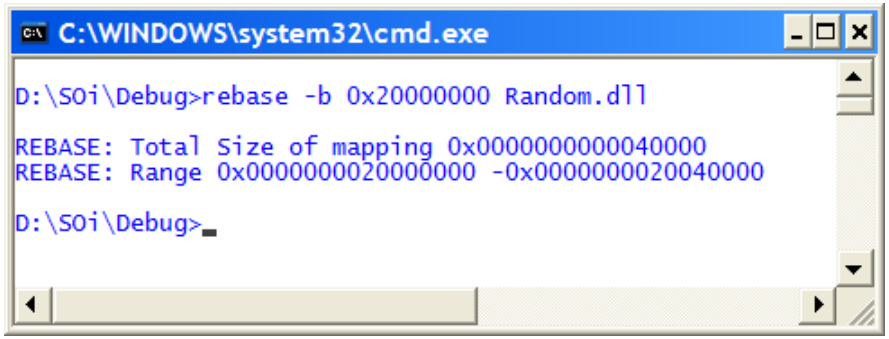
Rebase

Rebase – secção para realojamento de código - .reloc



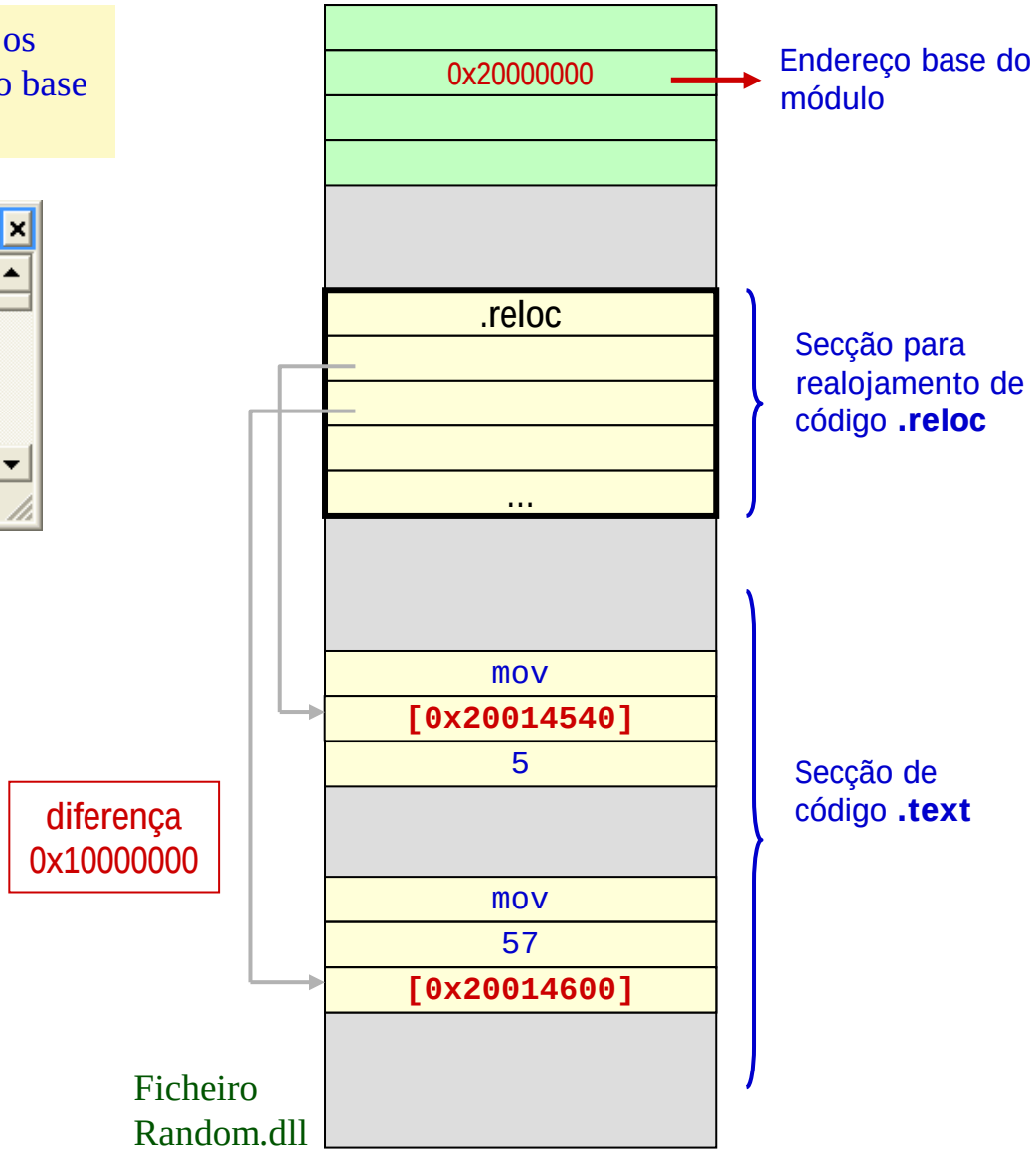
Rebase – *Rebase* da DLL Random.dll

O utilitário **rebase** vai proceder ao realojamento de todos os módulos indicados começando por atribuir como endereço base o indicado na opção **–b 0x<????>**

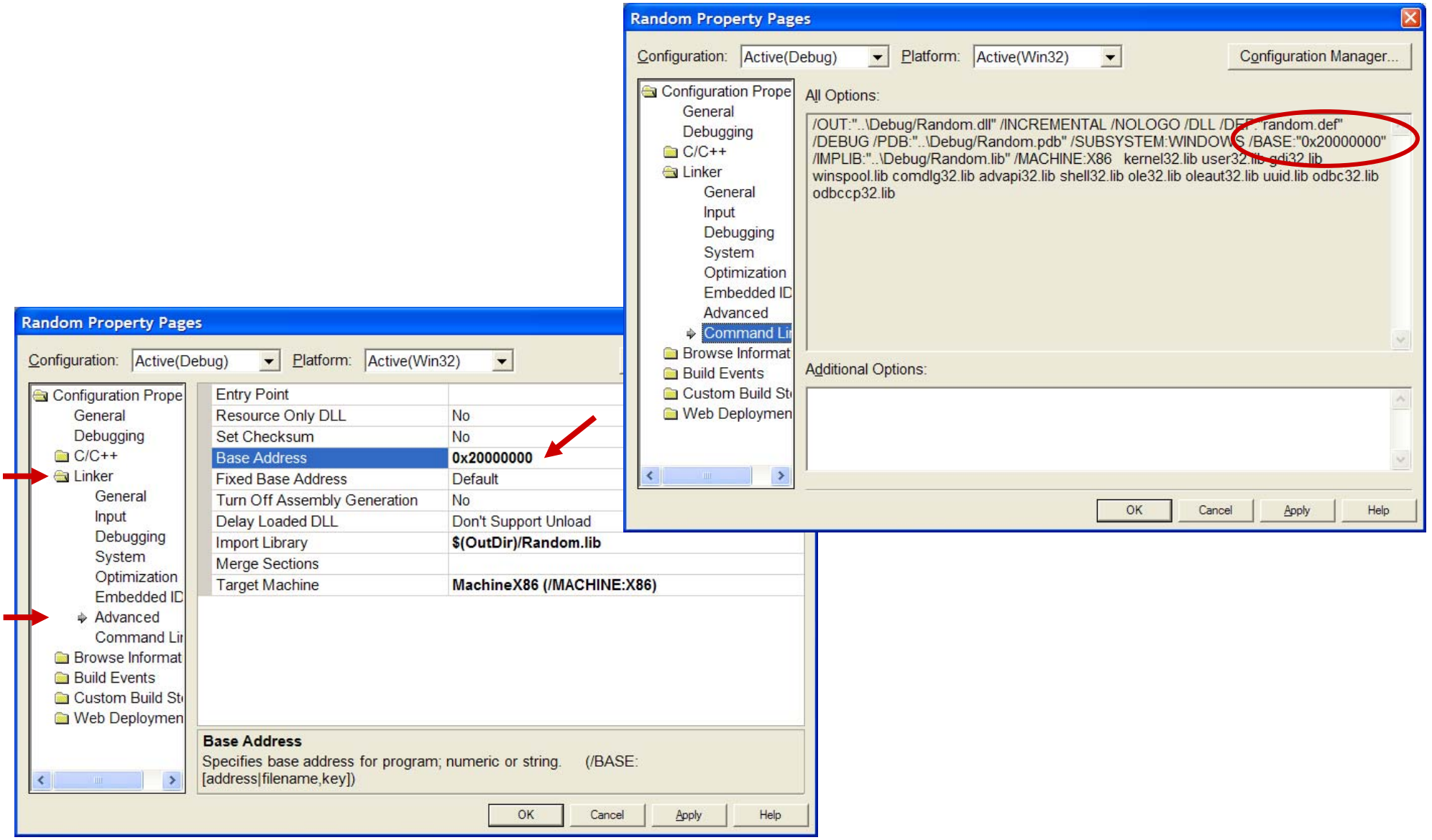


Neste exemplo vai apenas alterar o módulo **Random.dll** de forma a que o seu endereço base se altere para **0x20000000**. Assim executa os seguintes passos:

- Determina o endereço base existente no módulo e determina a diferença para o novo
- Altera endereço base do módulo
- Itera sobre a tabela de endereços da secção .reloc e adiciona essa diferença ao endereço guardado

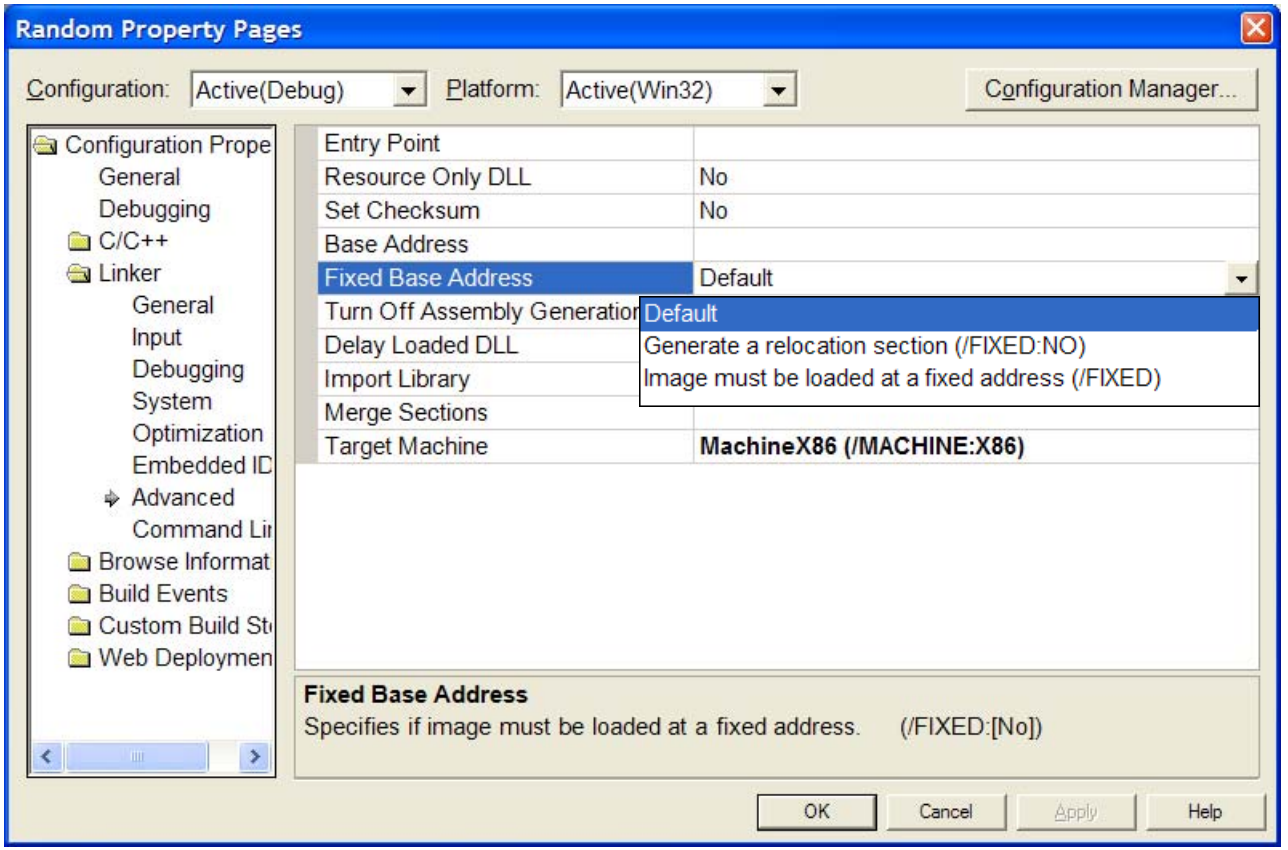


Rebase – Definição do endereço base no Visual Studio

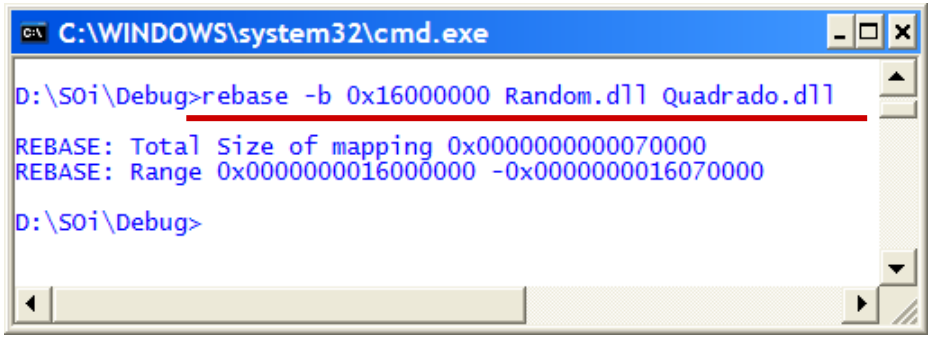


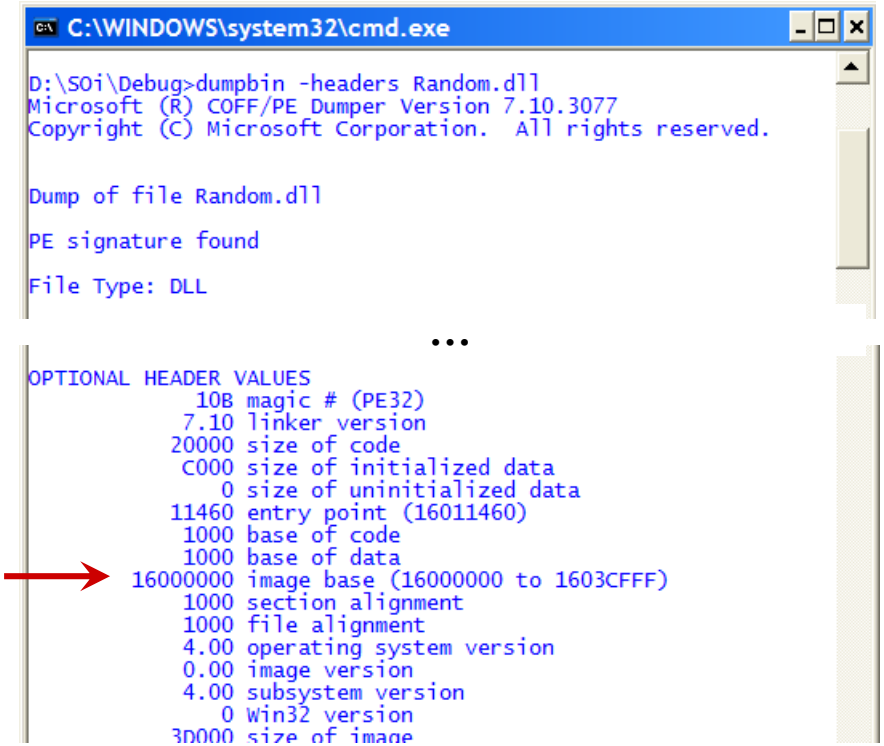
Rebase – Excluir a secção de realojamento de código da DLL

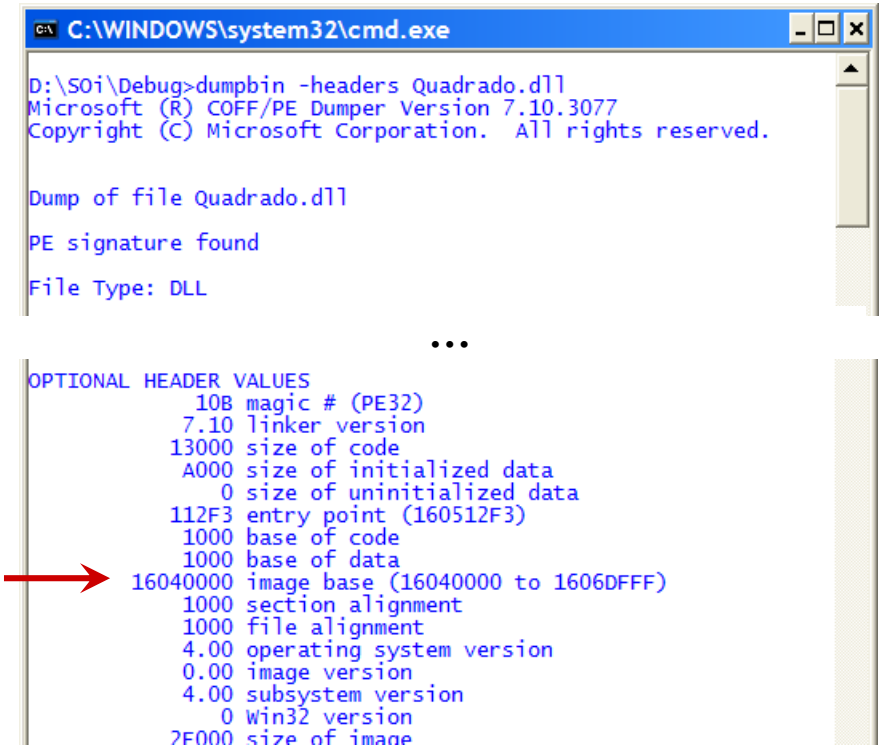
Existe a possibilidade de excluir a secção de realojamento de código da DLL através do **switch /FIXED do linker**. Neste caso o módulo só pode ser carregado no seu endereço base.
A imagem apresenta como pode ser definido este *switch* no Visual Studio.



Rebase – *Rebase* das DLL's Random.dll e Quadrado.dll





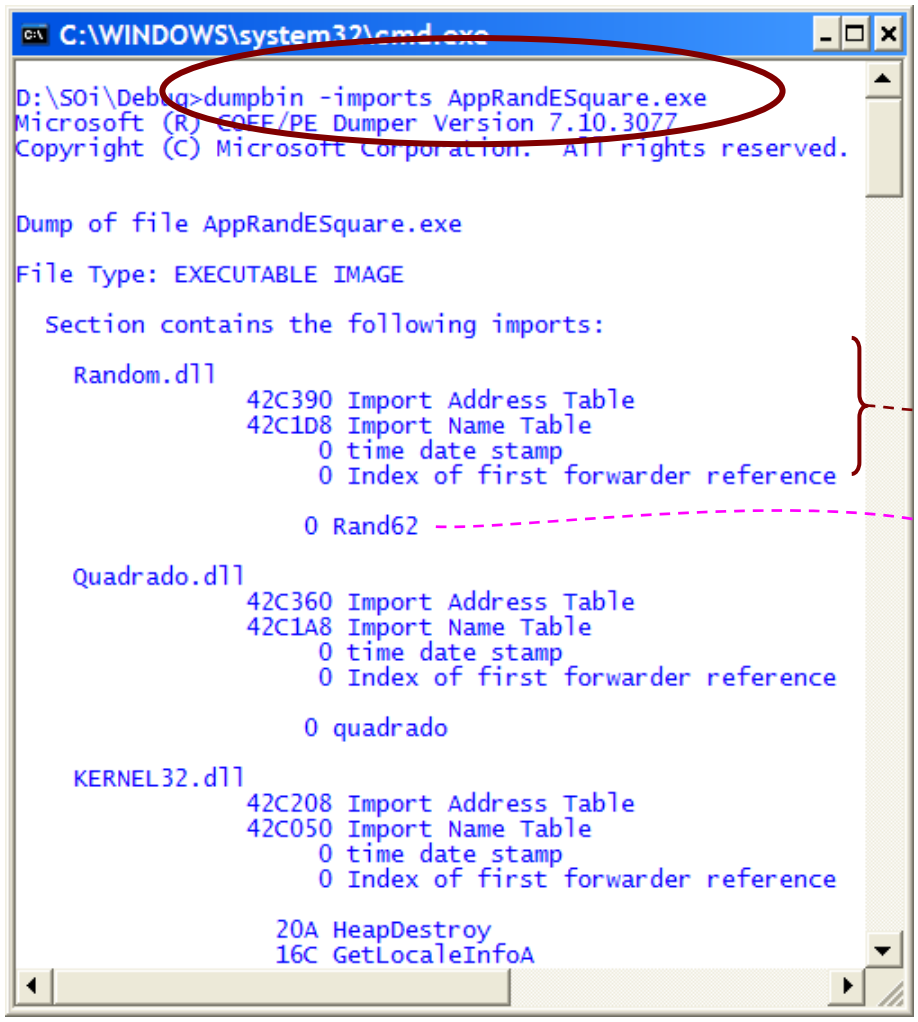


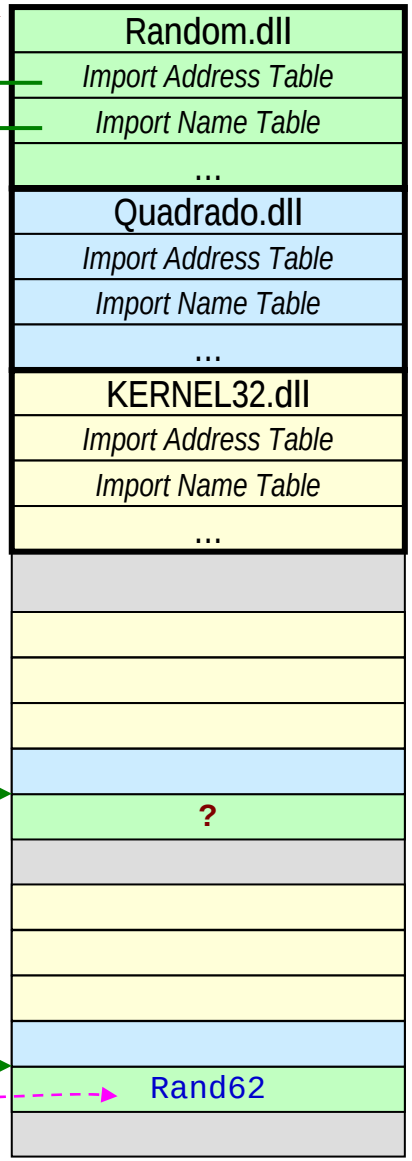
Rebase – Função da API

```
BOOL ReBaseImage(
  __in        PCSTR CurrentImageName,
  __in        PCSTR SymbolPath,
  __in        BOOL fReBase,
  __in        BOOL fRebaseSysfileOk,
  __in        BOOL fGoingDown,
  __in        ULONG CheckImageSize,
  __out       ULONG* OldImageSize,
  __out       ULONG_PTR* OldImageBase,
  __out       ULONG* NewImageSize,
  __in_out    ULONG_PTR* NewImageBase,
  __in        ULONG TimeStamp
);
```

Bind

Bind – Estrutura da secção de *import* de um módulo



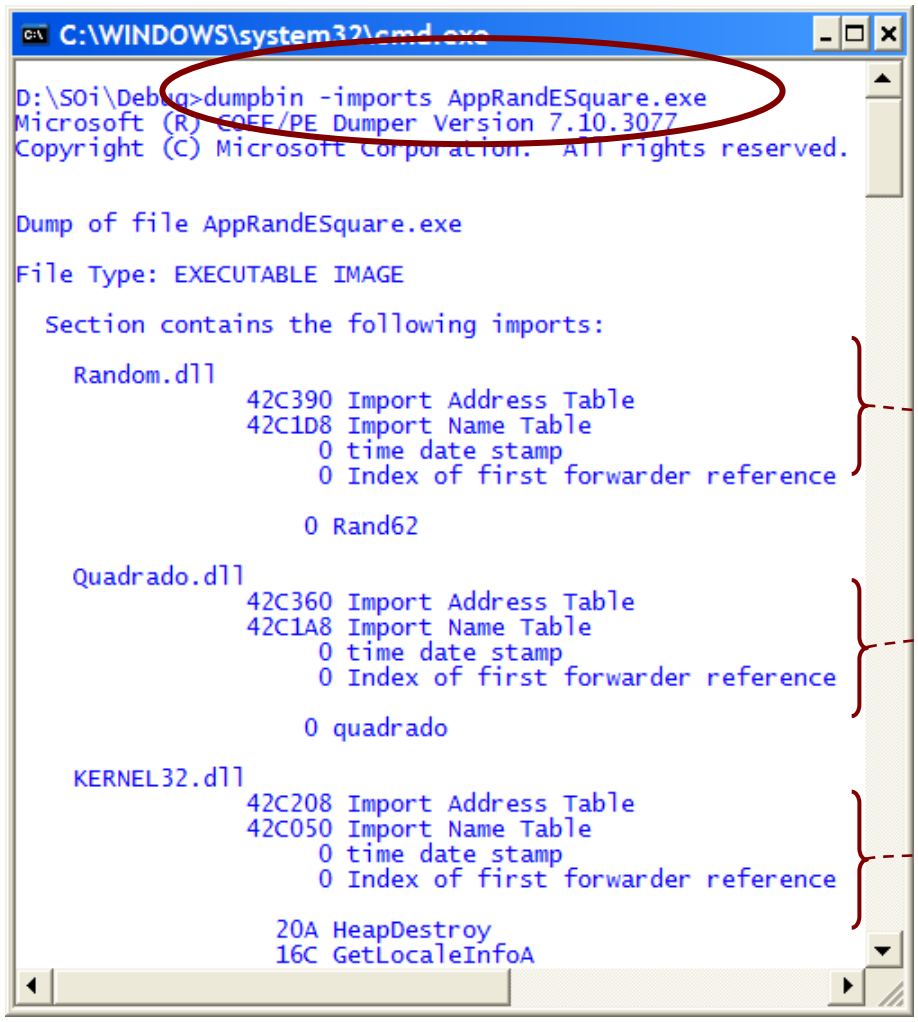


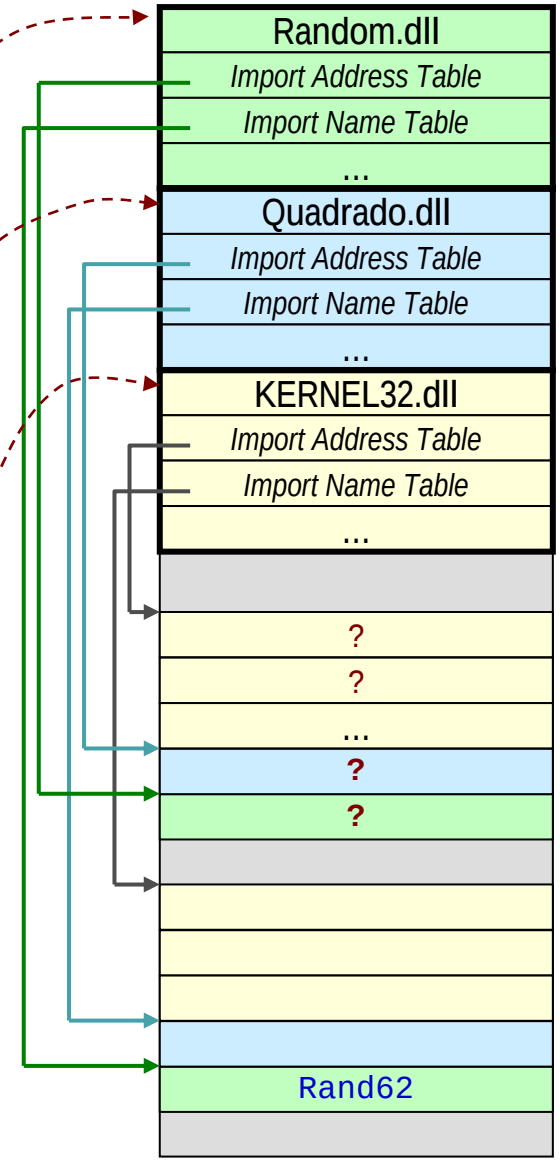
Secção *Import* de um Módulo binário (por exemplo um .EXE)

Import Address Tables

Import Name Tables

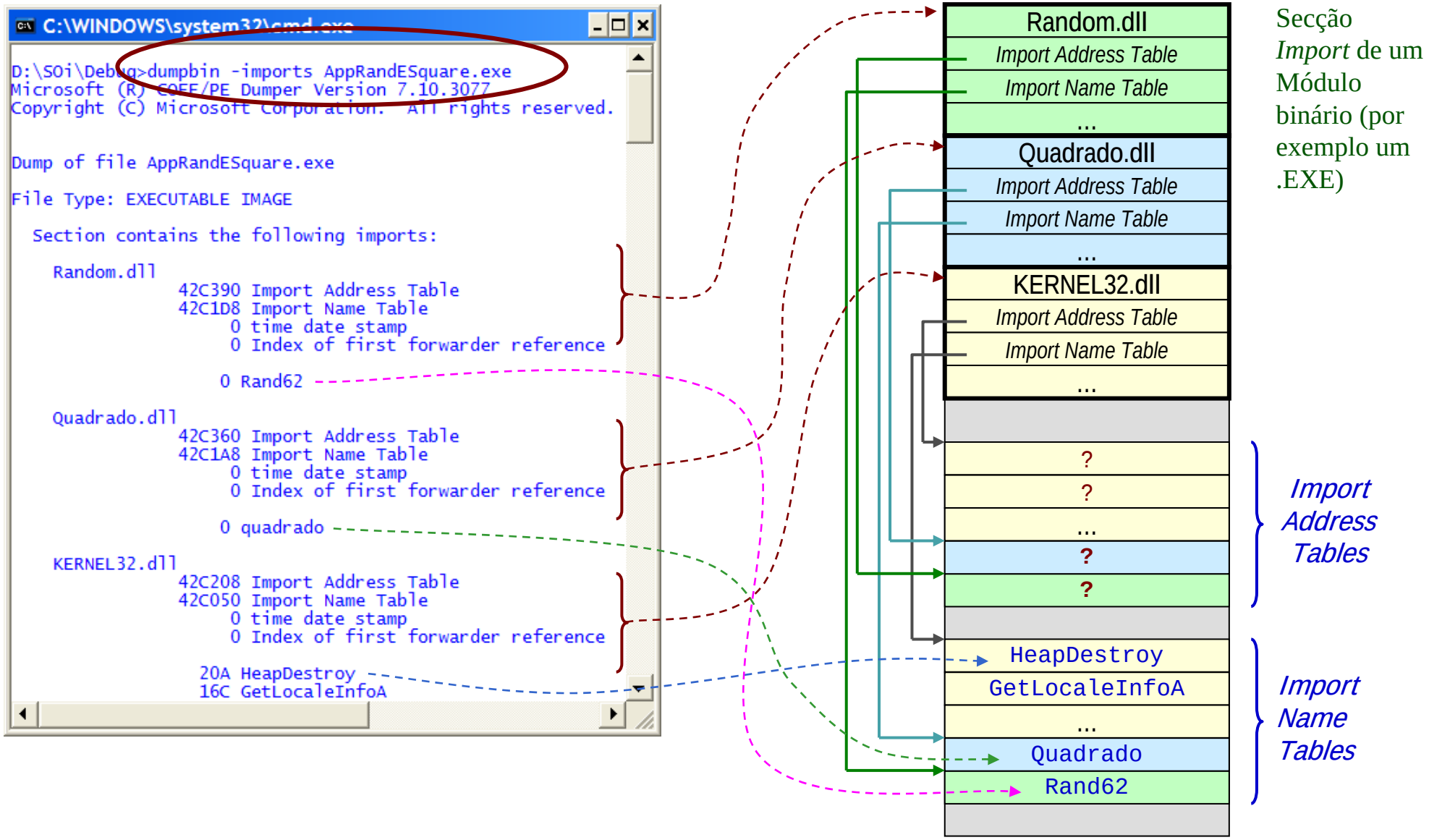
Bind – Estrutura da secção de *import* de um módulo





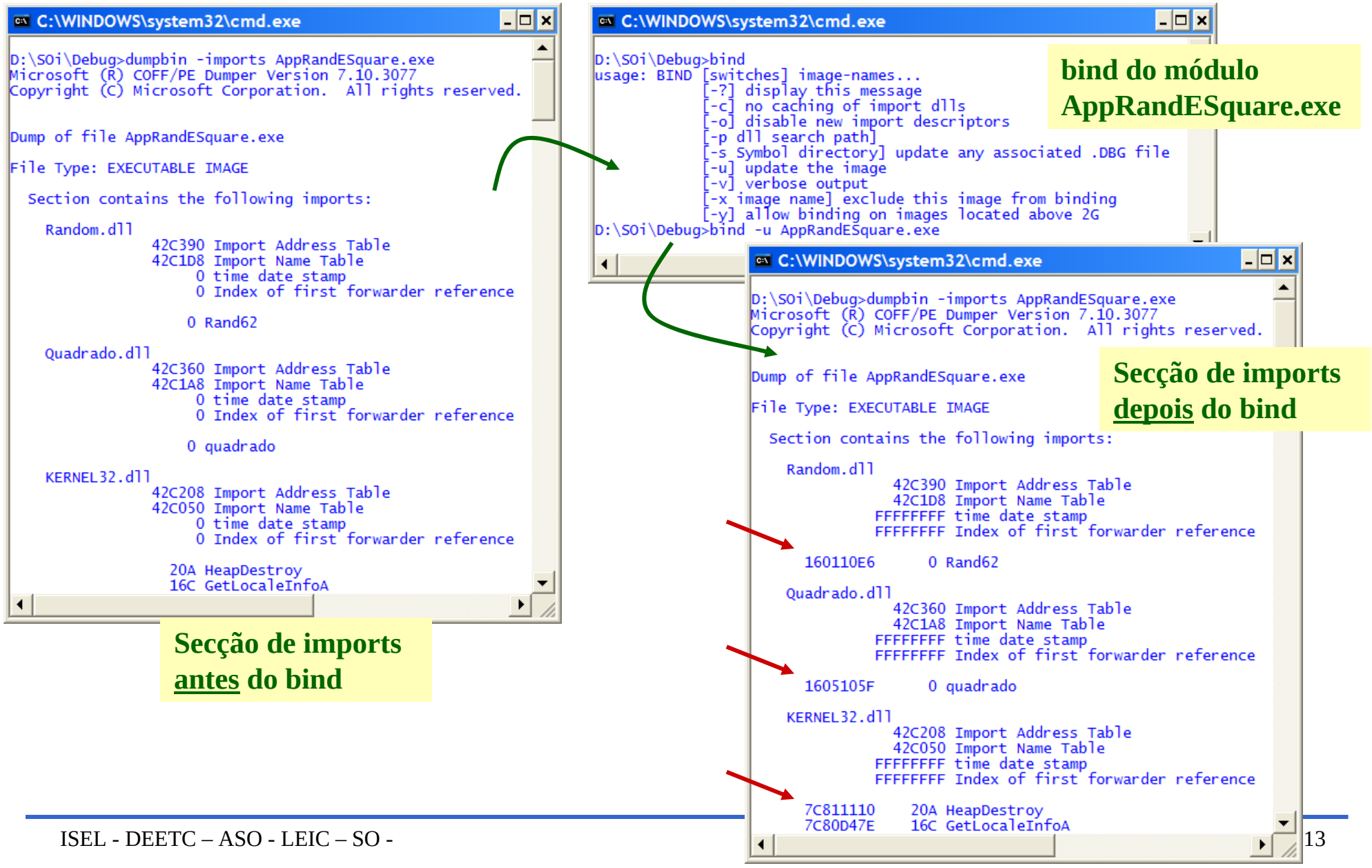
Secção *Import* de um Módulo binário (por exemplo um .EXE)

Bind – Estrutura da secção de *import* de um módulo



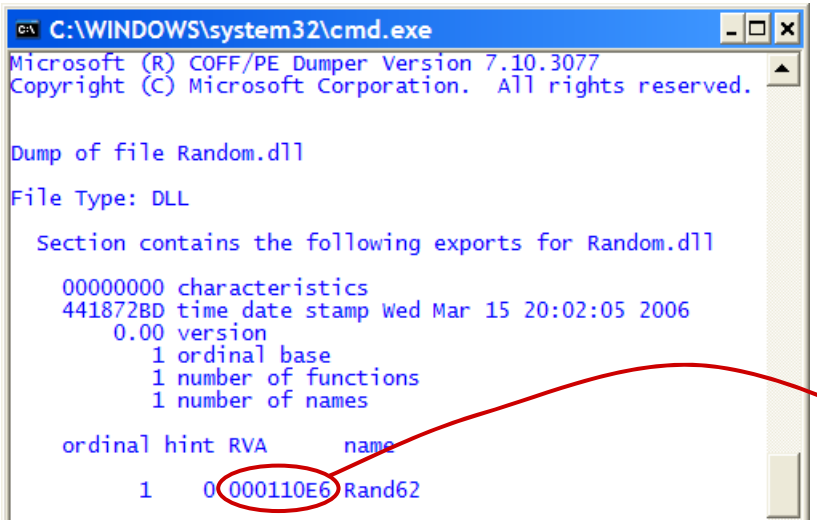
Secção *Import* de um Módulo binário (por exemplo um .EXE)

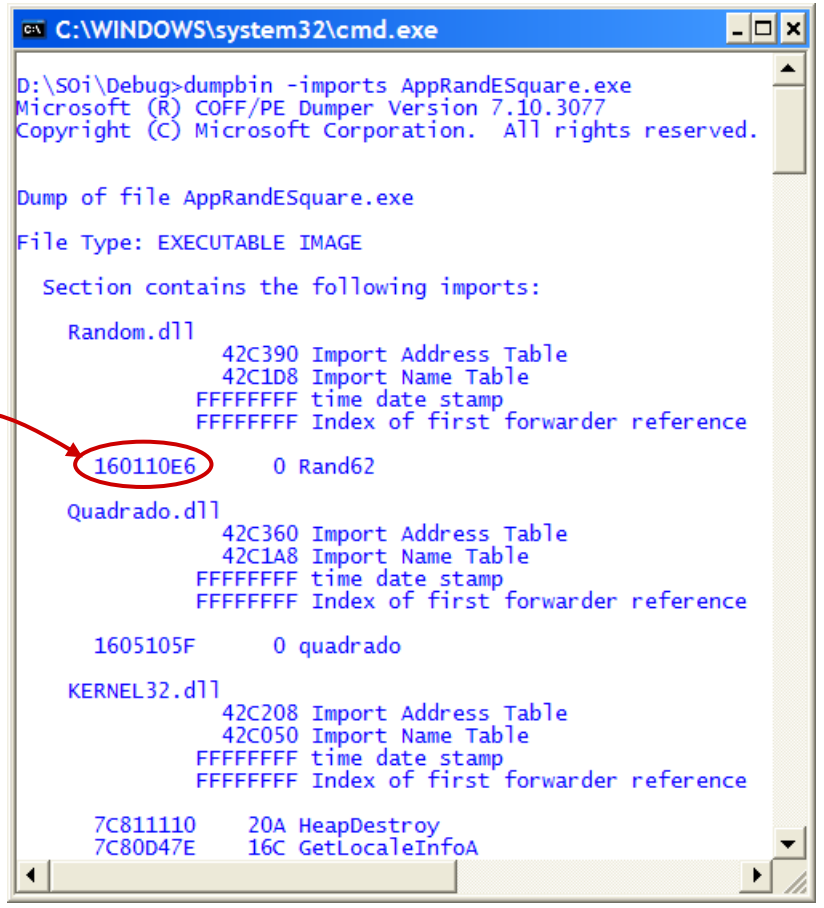
Bind – Diferenças na secção de *Imports* após realizado o *bind*



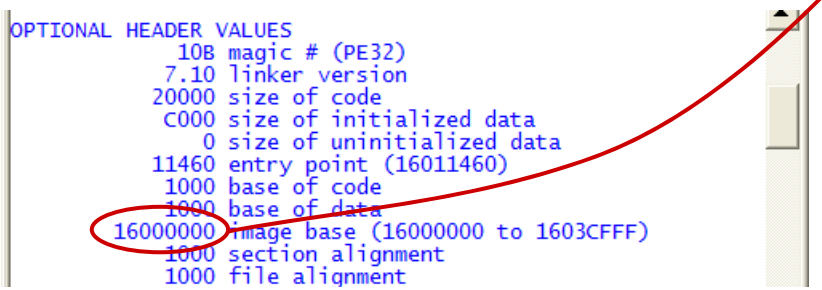
Bind – Como se obtém o novo valor associado a cada função

Da secção de *exports* do módulo Random.dll podemos encontrar o **RVA** (*Relative Virtual Address*) da função Rand62 que indica o offset da função no módulo DLL

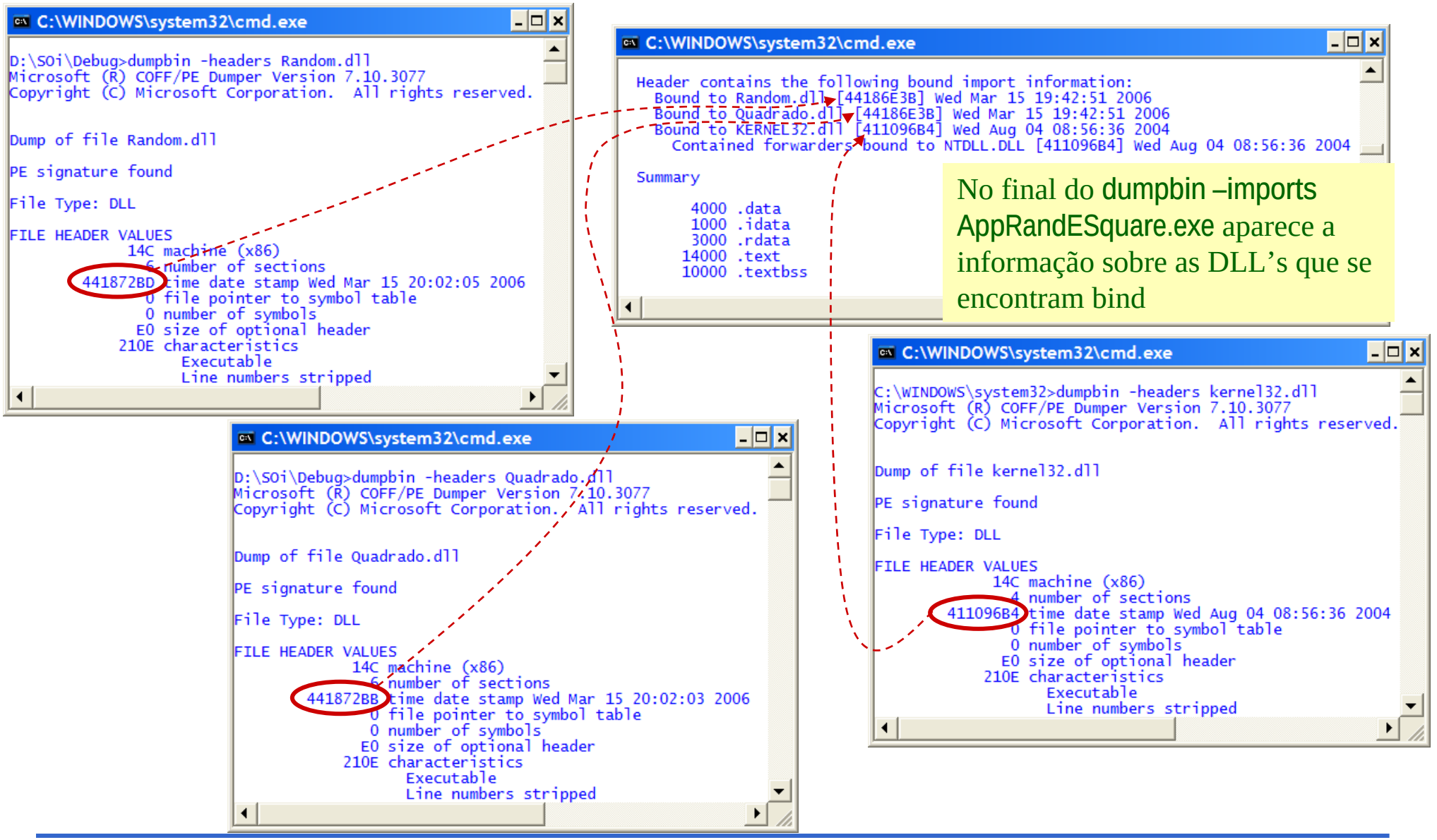




> dumpbin –headers Random.dll podemos verificar o endereço base do módulo



Bind – Cada módulo contém nos seus *headers* informação sobre os *binds*



Bind – Função da API

```
BOOL BindImageEx(
  __in        DWORD Flags,
  __in        PSTR ImageName,
  __in        PSTR DllPath,
  __in        PSTR SymbolPath,
  __in        PIMAGEHLP_STATUS_ROUTINE StatusRoutine
);
```