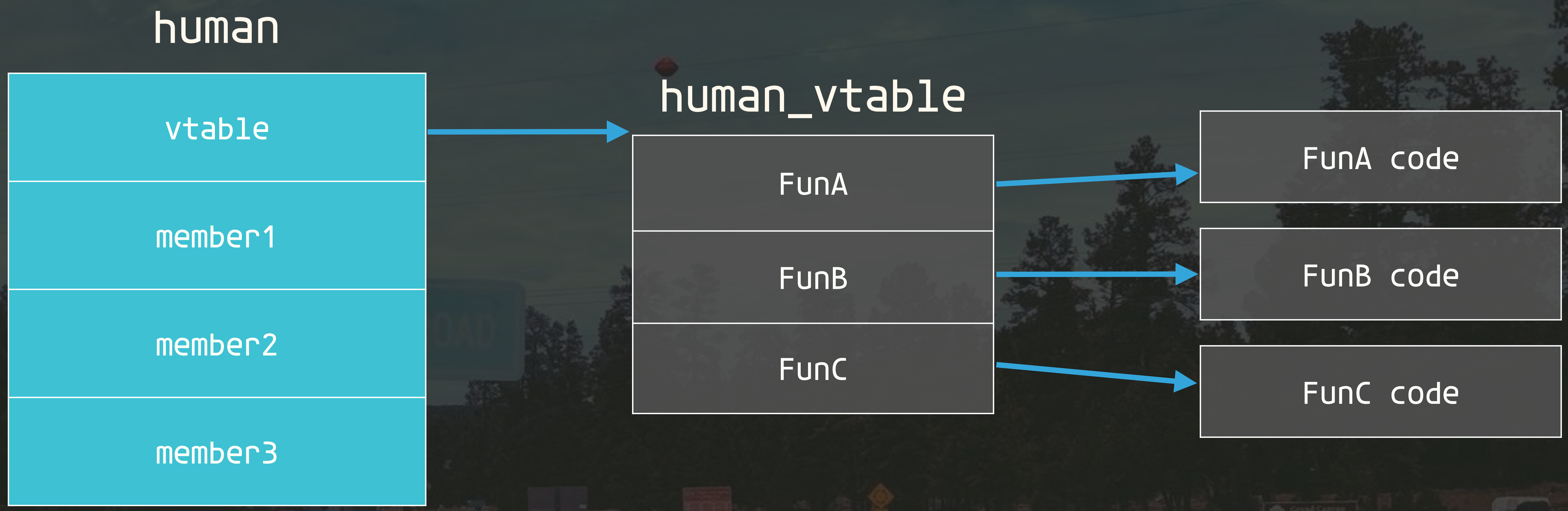
新竹阿婆天天逆向(工程)

terrynini38514

terrynini

LAST WEEK

繼承



human

vtable

member1

member2

member3

human_vtable

FunA

FunB

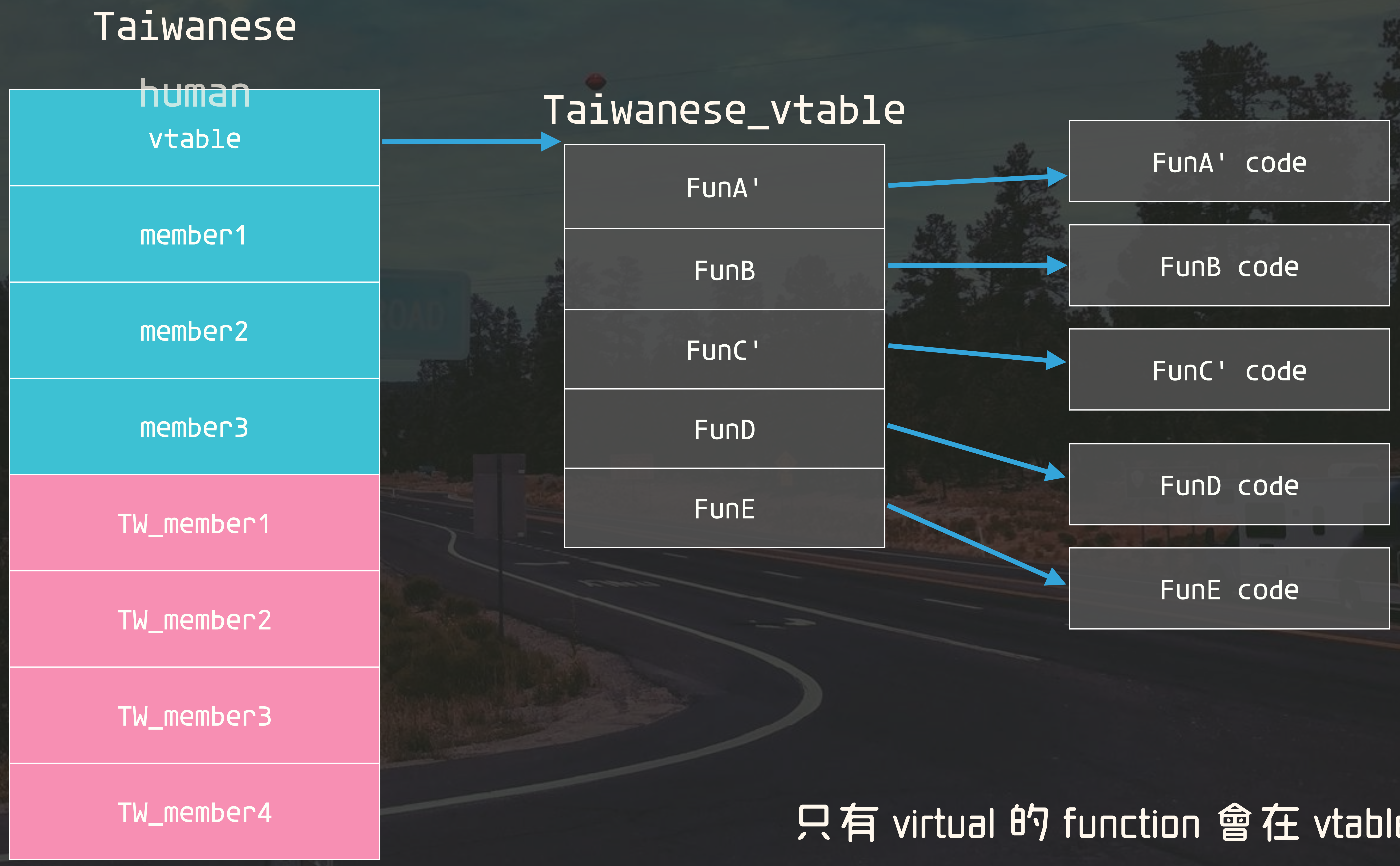
FunC

FunA code

FunB code

FunC code

繼承



只有 virtual 的 function 會在 vtable 中

且當使用指標或引用(&)才會真的去使用 vtable

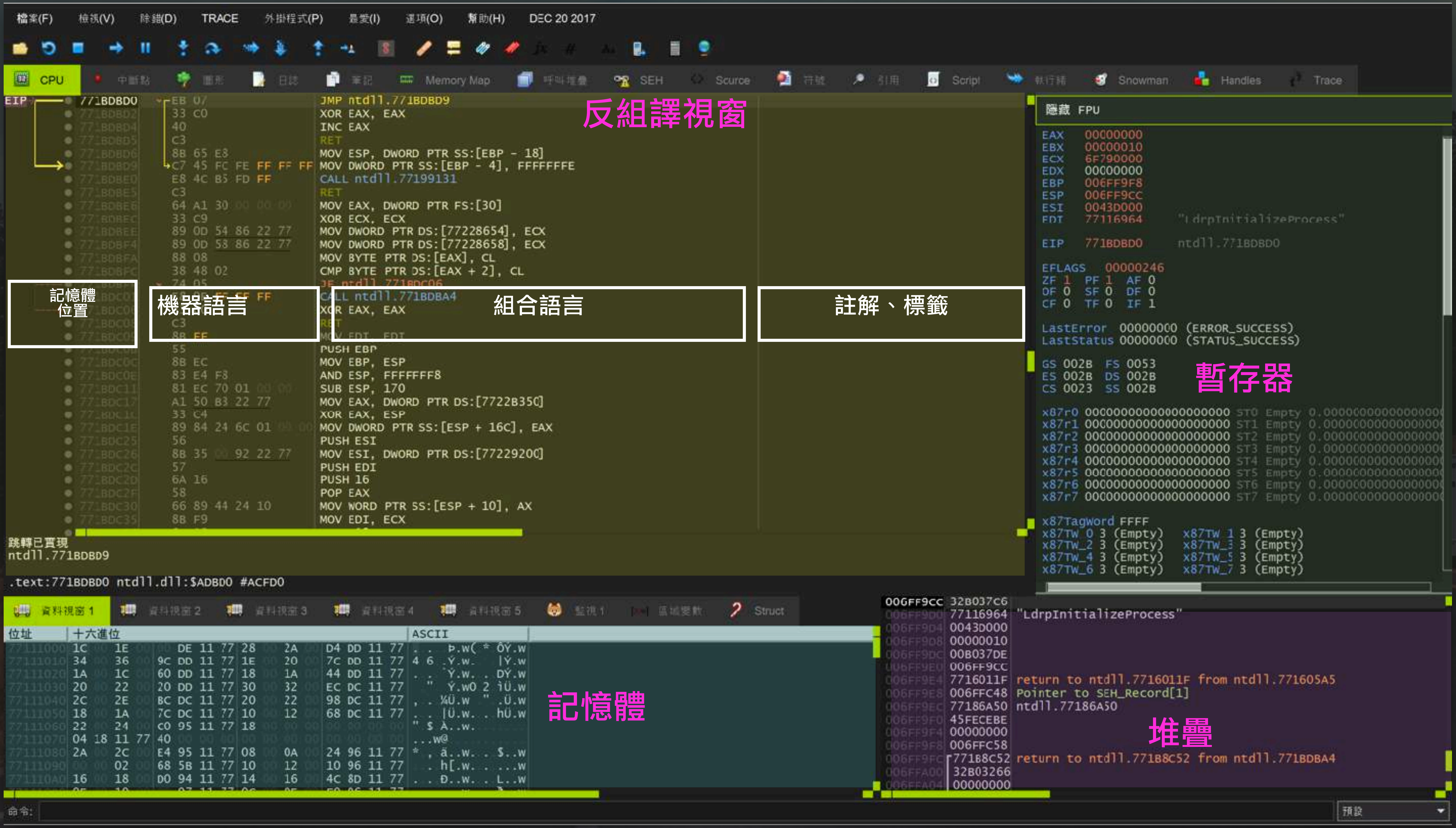
LAST WEEK

How to recover the member of a structure

Behavior

BASIC ANALYZE SKILL

x64dbg CheatSheet



```
EIP  771BDBD0  EB 07                 JMP ntdll.771BDBD9
     771BDBD2  33 C0                 XOR EAX, EAX
     771BDBD4  40                    INC EAX
     771BDBD5  C3                    RET
     771BDBD6  8B 65 E8              MOV ESP, DWORD PTR SS:[EBP - 18]
     771BDBD9  C7 45 FC FE FF FF FF  MOV DWORD PTR SS:[EBP - 4], FFFFFFFE
     771BDBE0  E8 4C B5 FD FF        CALL ntdll.77199131
     771BDBE5  C3                    RET
     771BDBE6  64 A1 30 00 00 00     MOV EAX, DWORD PTR FS:[30]
     771BDBEC  33 C9                 XOR ECX, ECX
     771BDBEE  89 0D 54 86 22 77     MOV DWORD PTR DS:[77228654], ECX
     771BDBF4  89 0D 58 86 22 77     MOV DWORD PTR DS:[77228658], ECX
     771BDBFA  88 08                 MOV BYTE PTR DS:[EAX], CL
     771BDBFC  38 48 02              CMP BYTE PTR DS:[EAX + 2], CL
               74 05                 JE ntdll.771BDC06
       BDC01            FF FF FF     CALL ntdll.771BDBA4
       BDC06                         XOR EAX, EAX
       BDC08   C3                    RET
       BDC09   8B FF                 MOV EDI, EDI
               55                    PUSH EBP
     771BDC0C  8B EC                 MOV EBP, ESP
     771BDC0E  83 E4 F8              AND ESP, FFFFFFF8
     771BDC11  81 EC 70 01 00 00     SUB ESP, 170
     771BDC17  A1 50 B3 22 77        MOV EAX, DWORD PTR DS:[7722B350]
     771BDC1C  33 C4                 XOR EAX, ESP
     771BDC1E  89 84 24 6C 01 00 00  MOV DWORD PTR SS:[ESP + 16C], EAX
     771BDC25  56                    PUSH ESI
     771BDC26  8B 35 00 92 22 77     MOV ESI, DWORD PTR DS:[77229200]
     771BDC2C  57                    PUSH EDI
     771BDC2D  6A 16                 PUSH 16
     771BDC2F  58                    POP EAX
     771BDC30  66 89 44 24 10        MOV WORD PTR SS:[ESP + 10], AX
     771BDC35  8B F9                 MOV EDI, ECX
跳轉已實現
ntdll.771BDBD9

.text:771BDBD0 ntdll.dll:$ADBD0 #ACFD0
```

```
EAX     00000000
EBX     00000010
ECX     6F790000
EDX     00000000
EBP     006FF9F8
ESP     006FF9CC
ESI     0043D000
EDI     77116964      "LdrpInitializeProcess"

EIP     771BDBD0      ntdll.771BDBD0

EFLAGS    00000246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus 00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

x87r0 00000000000000000000 ST0 Empty 0.000000000000000000
x87r1 00000000000000000000 ST1 Empty 0.000000000000000000
x87r2 00000000000000000000 ST2 Empty 0.000000000000000000
x87r3 00000000000000000000 ST3 Empty 0.000000000000000000
x87r4 00000000000000000000 ST4 Empty 0.000000000000000000
x87r5 00000000000000000000 ST5 Empty 0.000000000000000000
x87r6 00000000000000000000 ST6 Empty 0.000000000000000000
x87r7 00000000000000000000 ST7 Empty 0.000000000000000000

x87TagWord FFFF
x87TW_0 3 (Empty)   x87TW_1 3 (Empty)
x87TW_2 3 (Empty)   x87TW_3 3 (Empty)
x87TW_4 3 (Empty)   x87TW_5 3 (Empty)
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)
```

```
位址      十六進位                                          ASCII
77111000  1C 00 1E 00 00 DE 11 77 28 00 2A 00 D4 DD 11 77  . . Þ.w( * ÔÝ.w
77111010  34 00 36 00 9C DD 11 77 1E 00 20 00 7C DD 11 77  4 6 .Ý.w.   |Ý.w
77111020  1A 00 1C 00 60 DD 11 77 18 00 1A 00 44 DD 11 77  . . `Ý.w. . DÝ.w
77111030  20 00 22 00 20 DD 11 77 30 00 32 00 EC DC 11 77    "  Ý.w0 2 ìÜ.w
77111040  2C 00 2E 00 BC DC 11 77 20 00 22 00 98 DC 11 77  , . ¼Ü.w  " .Ü.w
77111050  18 00 1A 00 7C DC 11 77 10 00 12 00 68 DC 11 77  . . |Ü.w. . hÜ.w
77111060  22 00 24 00 C0 95 11 77 18 00 00 00 00 00 00 00  " $ À..w........
77111070  04 18 11 77 40 00 00 00 00 00 00 00 00 00 00 00  ...w@...........
77111080  2A 00 2C 00 E4 95 11 77 08 00 0A 00 24 96 11 77  * , ä..w. . $..w
77111090  00 00 02 00 68 5B 11 77 10 00 12 00 10 96 11 77    . h[.w. . ...w
771110A0  16 00 18 00 D0 94 11 77 14 00 16 00 4C 8D 11 77  . . Ð..w. . L..w
```

```
006FF9CC  32B037C6
006FF9D0  77116964  "LdrpInitializeProcess"
006FF9D4  0043D000
006FF9D8  00000010
006FF9DC  00B037DE
006FF9E0  006FF9CC
006FF9E4  7716011F  return to ntdll.7716011F from ntdll.771605A5
006FF9E8  006FFC48  Pointer to SEH_Record[1]
006FF9EC  77186A50  ntdll.77186A50
006FF9F0  45FECEBE
006FF9F4  00000000
006FF9F8  006FFC58
006FF9FC  771B8C52  return to ntdll.771B8C52 from ntdll.771BDBA4
006FFA00  32B03266
006FFA04  00000000
```

x64dbg CheatSheet

按鍵	功能	按鍵	功能
F4	執行到指定的行為止	Ctrl+G	跳到某個address
F7	單步執行(Step into)	Enter	查看Function
F8	單步執行(Step over)	*	回到EIP的位置
F9	執行	-/+	回到上/下一個位置
Ctrl+F2	重新開始	;/:	新增註解/標籤
Ctrl+F9	執行到return後停止	f2	下斷點
alt+C	disassemble	alt+G	Control flow graph

LAB

分析策略

- 套用已知模型
- Magic number
- 預測程式碼
- 抽象化！

PE FILE FORMAT

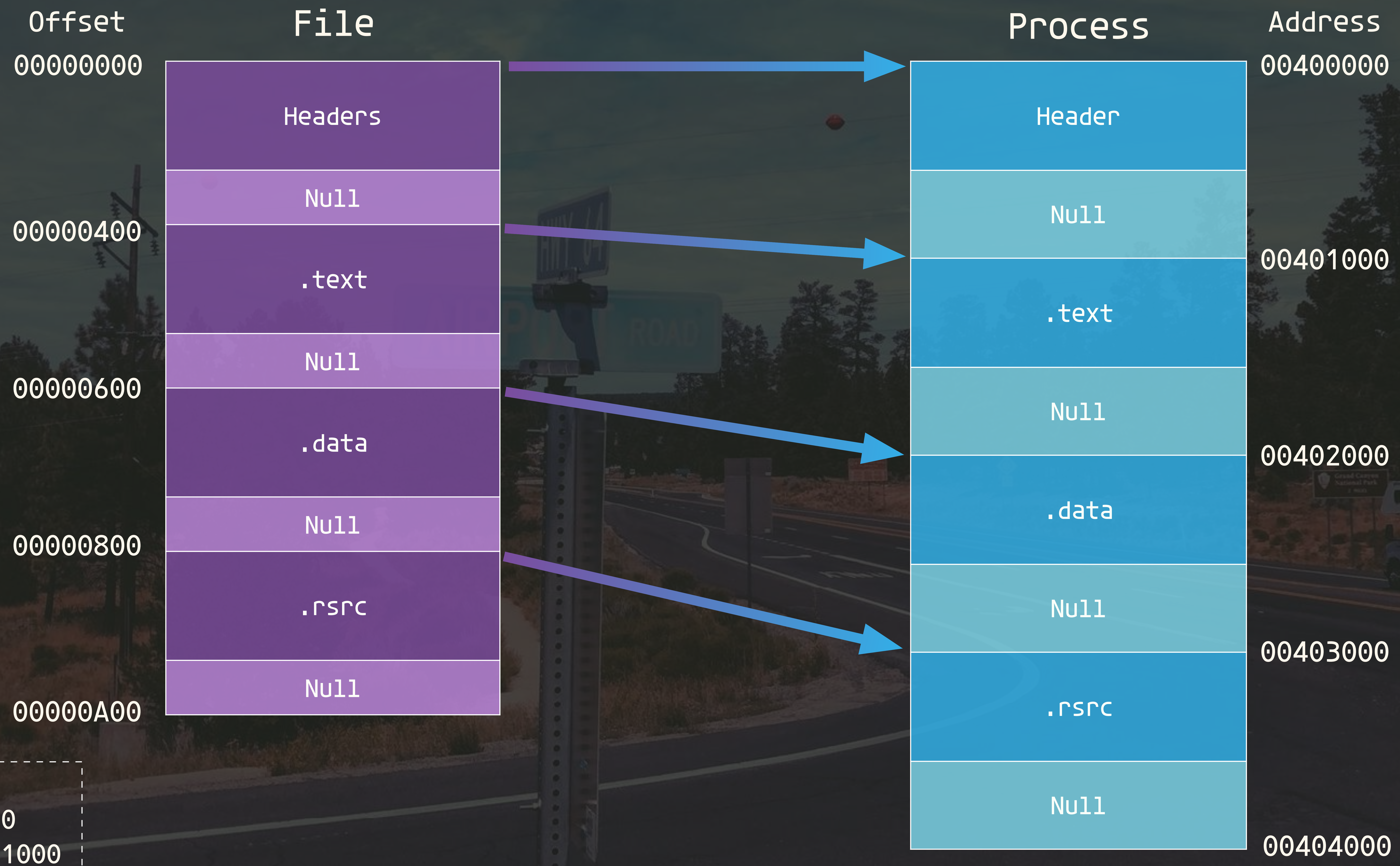
File format

- Windows 使用 PE(Portable Executable) 作為 executable、DLL、Driver 的格式
    - 32 位元的版本稱作 PE 或 PE32
    - 64 位元的版本稱作 PE+ 或是 PE32+
- Mac OS X 使用 Mach-O
- Linux 及 Unix 使用 ELF (Executable Linkable Format)
  (try this : file /boot/efi/EFI/ubuntu/grubx64.efi)

Some PE viewer

- 010 editor
- PE Bear

PE file format

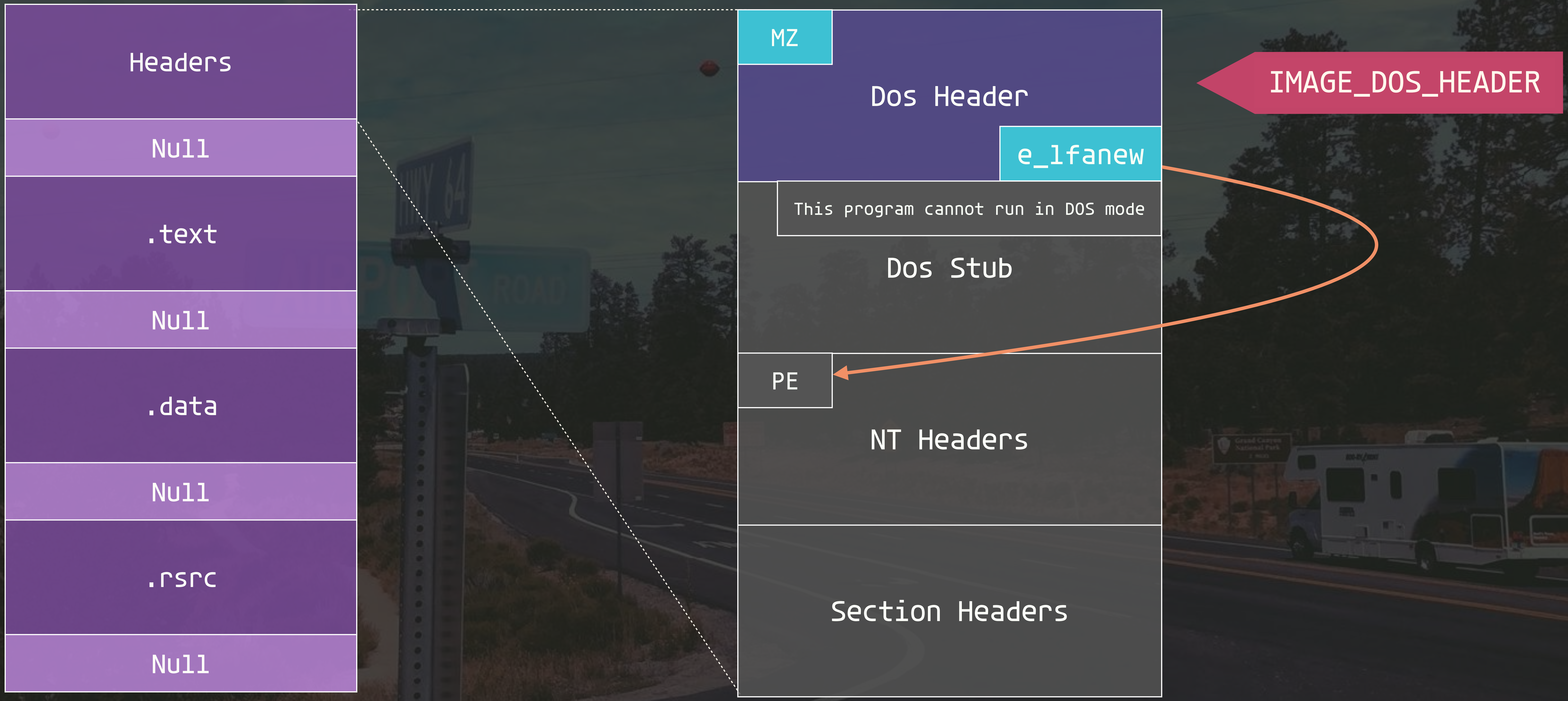


Offset	File
00000000	Headers
	Null
00000400	.text
	Null
00000600	.data
	Null
00000800	.rsrc
	Null
00000A00	

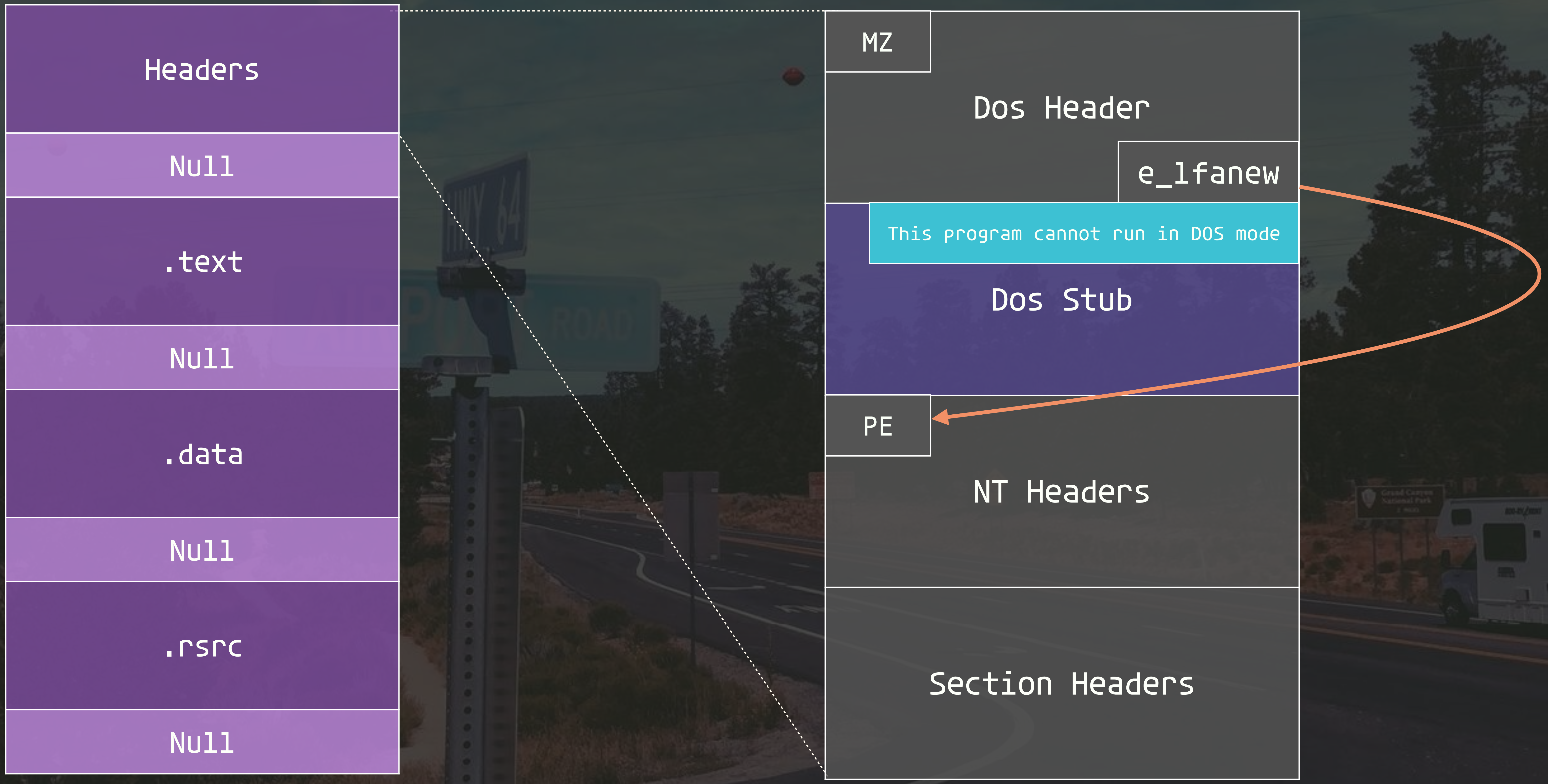
Process	Address
Header	00400000
Null	
.text	00401000
Null	
.data	00402000
Null	
.rsrc	00403000
Null	
	00404000

```
NumberOfSections : 3
FileAlignment : 0x200
SectionAlignment: 0x1000
```

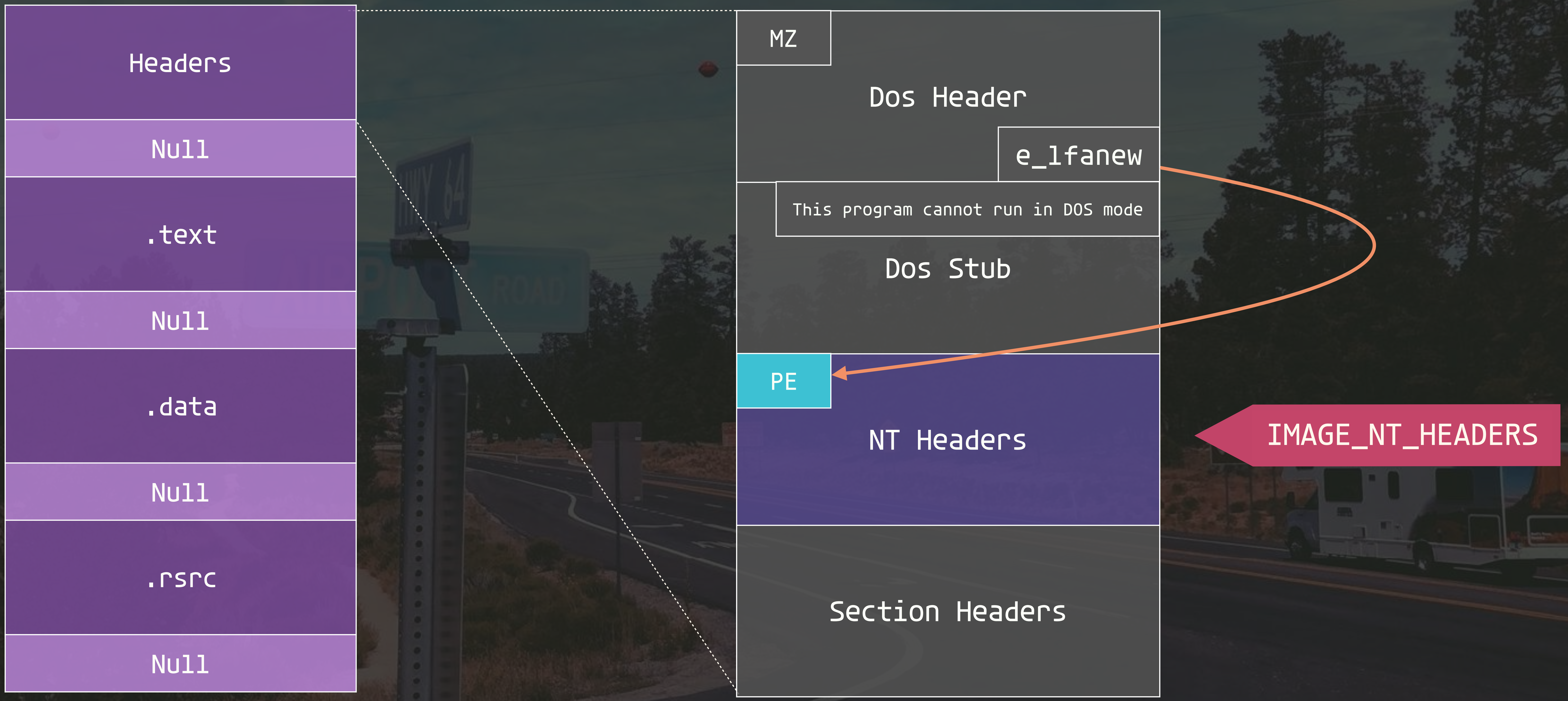
PE file format



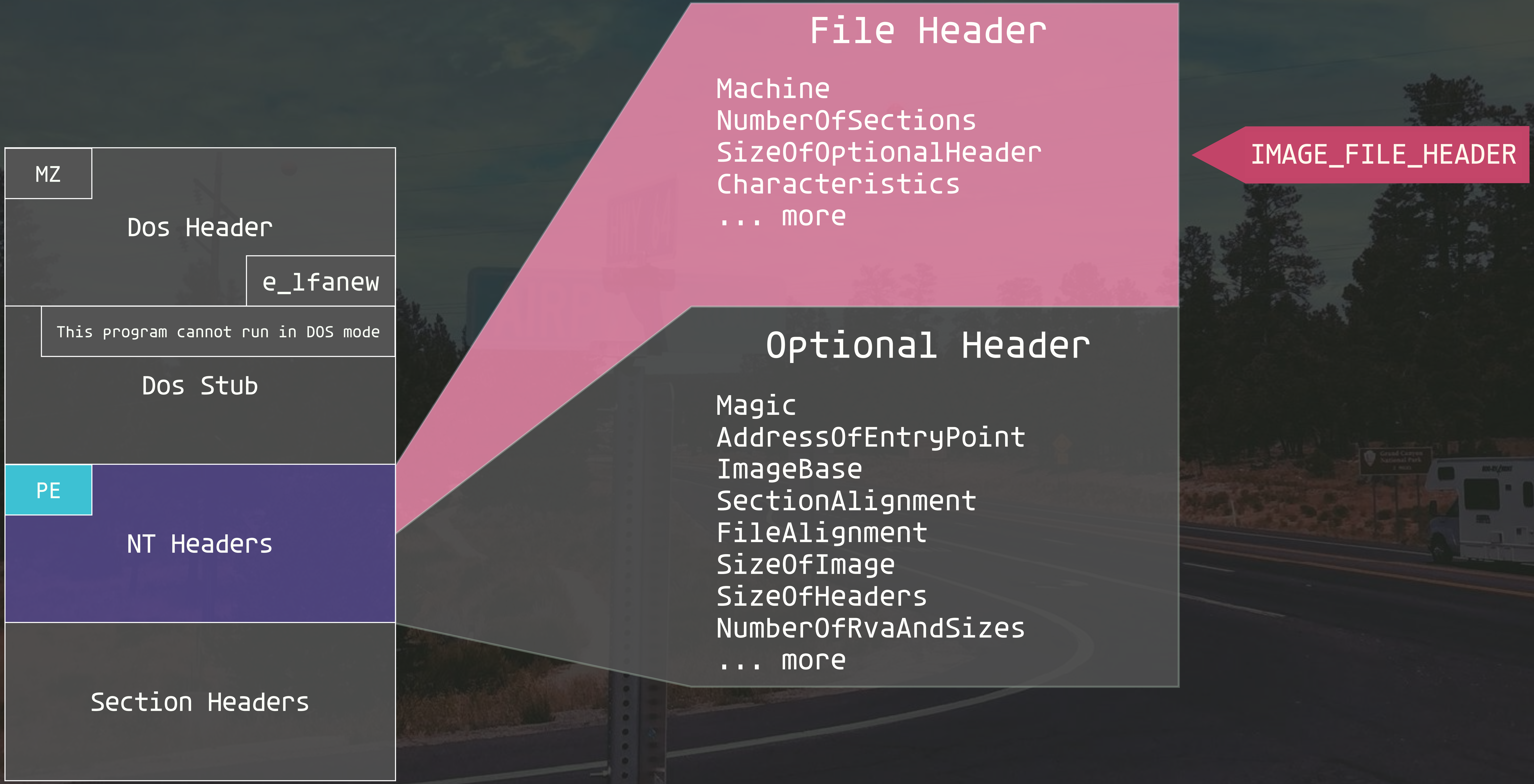
PE file format



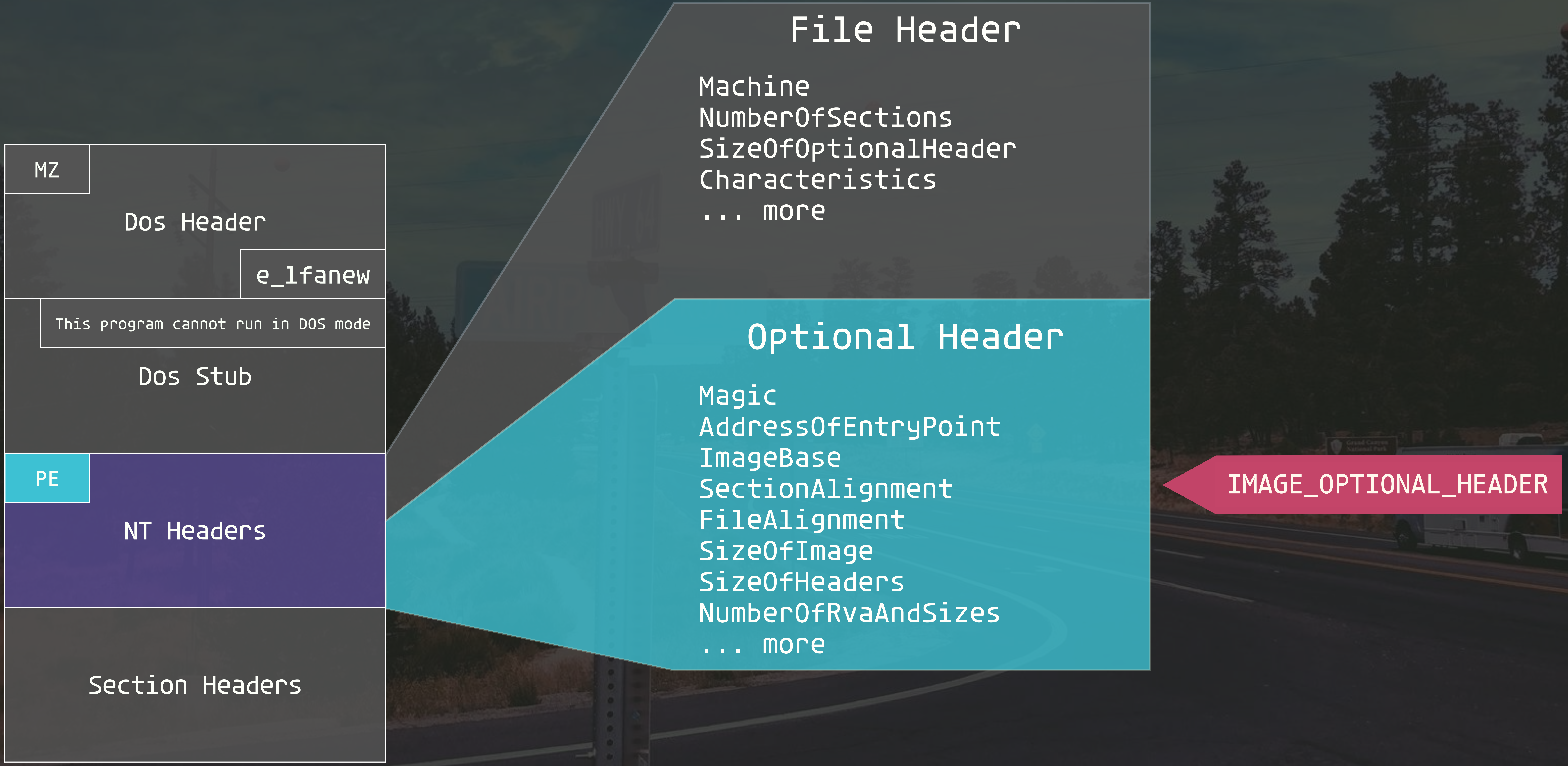
PE file format



PE file format



PE file format

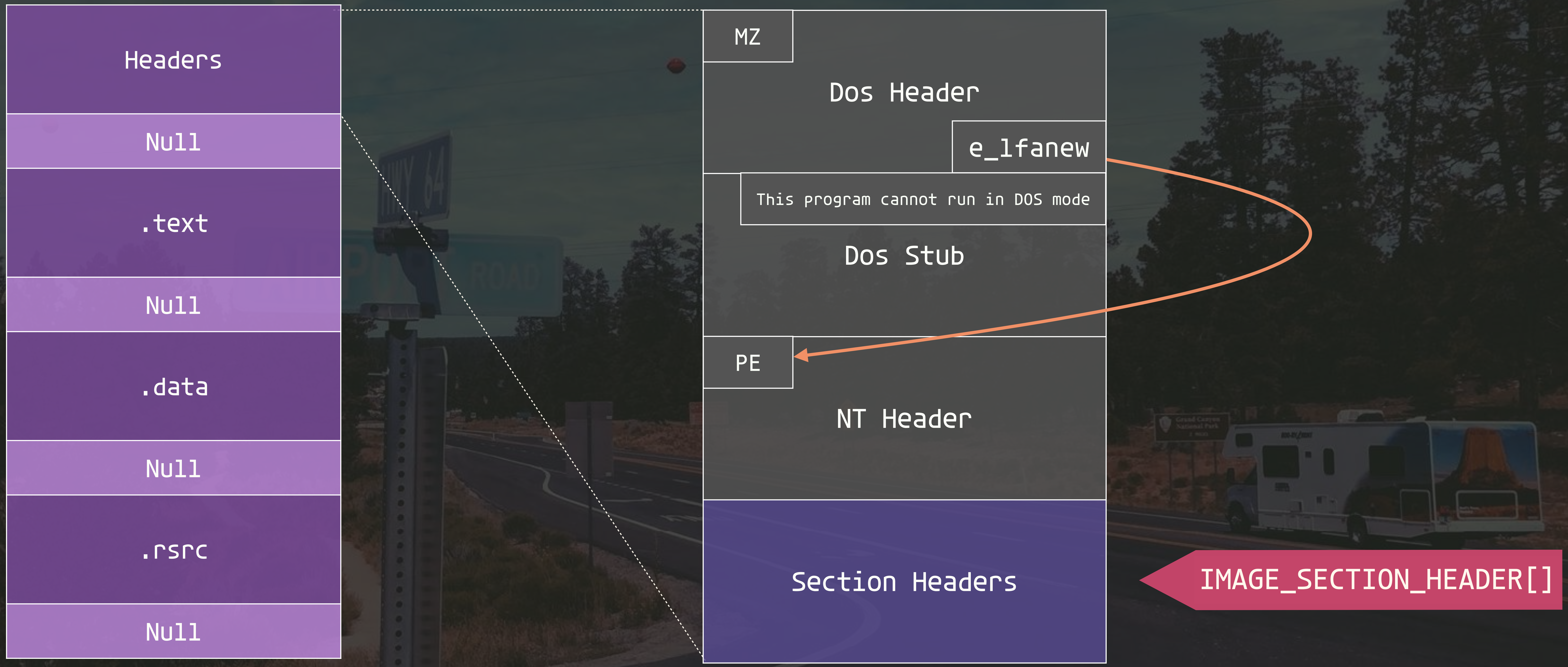


IMAGE_OPTIONAL_HEADER

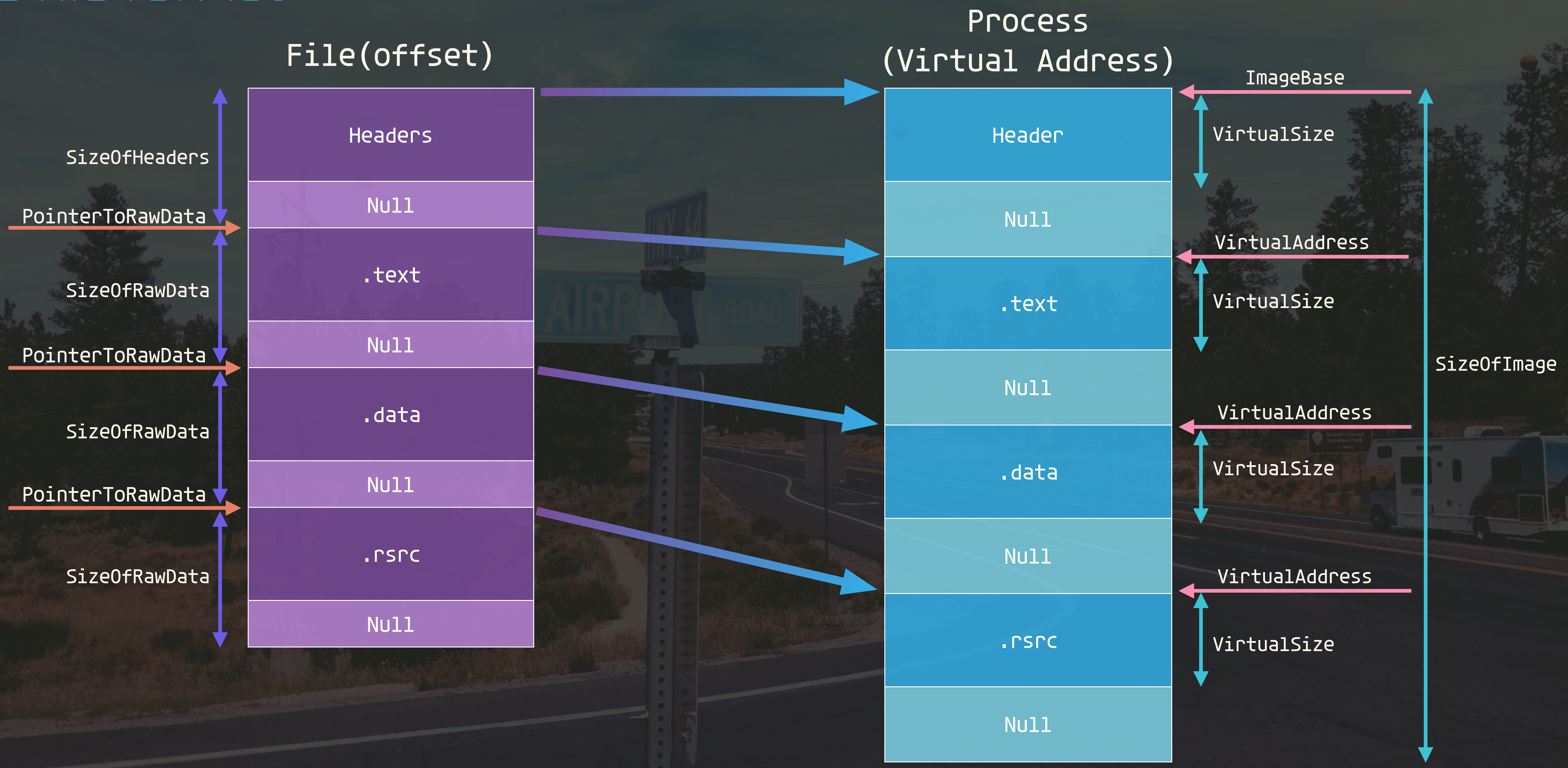
```
DataDirectory[0] = Export Directory
DataDirectory[1] = Import Directory
DataDirectory[2] = Resource Directory
DataDirectory[3] = Exception Directory
DataDirectory[4] = Security Directory
DataDirectory[5] = Base Relocation Table
DataDirectory[6] = Debug Directory
DataDirectory[7] = Architecture Specific Data
DataDirectory[8] = RVA of GlobalPtr
DataDirectory[9] = TLS Directory
DataDirectory[10] = Load Configuration Directory
DataDirectory[11] = Bound Import Directory
DataDirectory[12] = Import Address Table
DataDirectory[13] = Delay Load Import Descriptors
DataDirectory[14] = .NET header
DataDirectory[15] = Reversed Directory
```

IMAGE_DATA_DIRECTORY

PE file format



PE file format

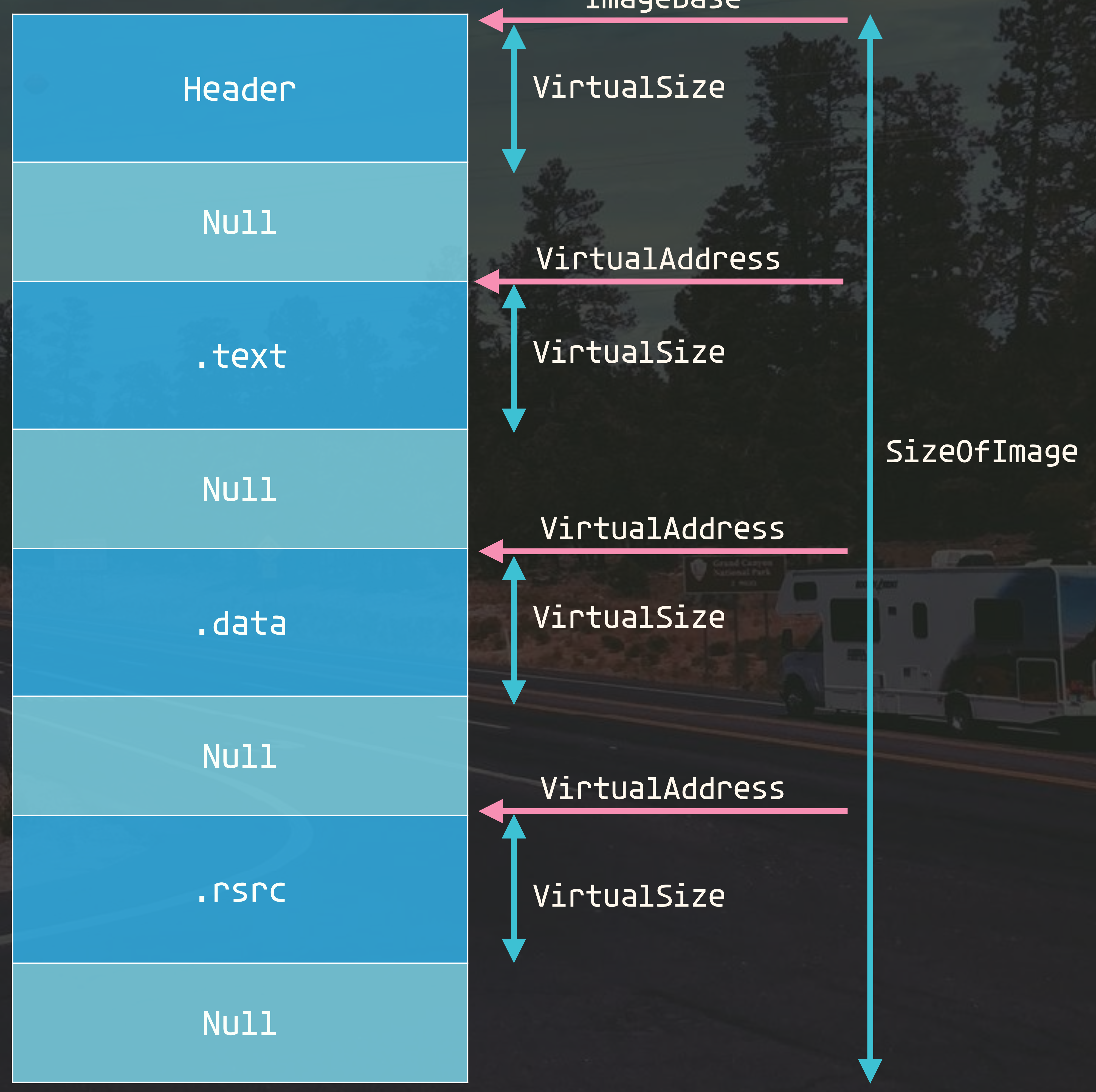


孔明の罠

PE 這裡的 Virtual Address
其實是 Relative Virtual Address

VA = RVA + ImageBase

Process
(Virtual Address)

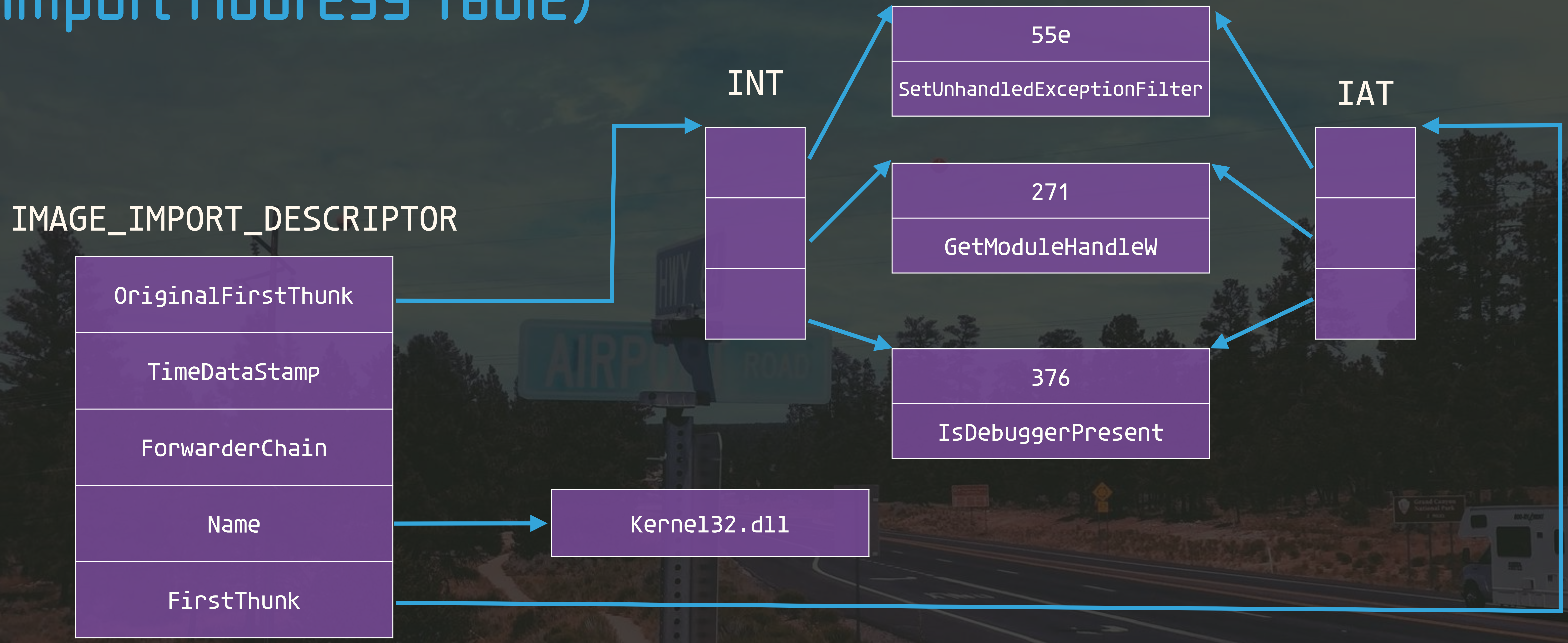


Sections

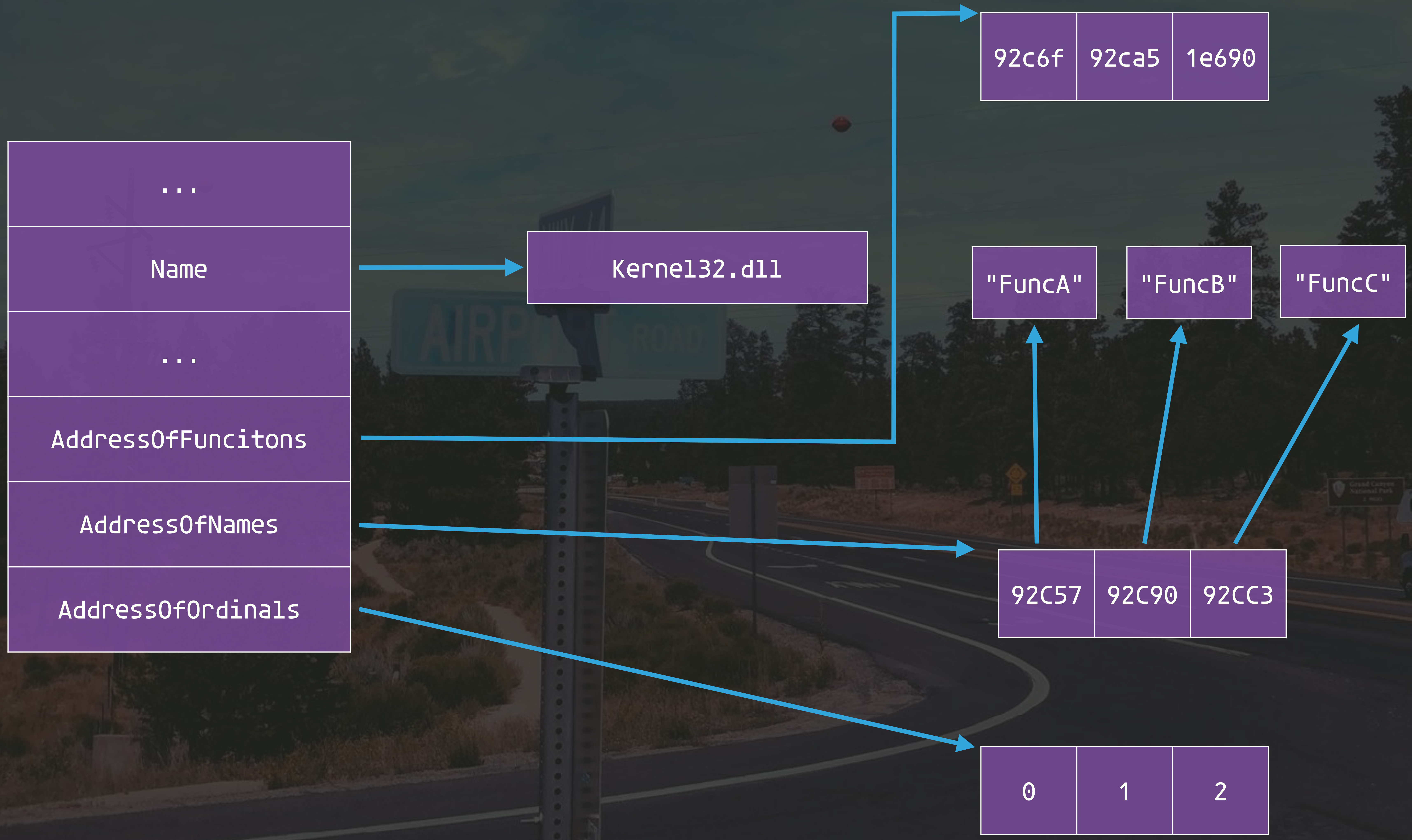
- .text 程式碼
- .data 放 data 的地方
- .rdata 唯讀的 data
- .bss 沒初始化的全域或靜態變數

- .idata 跟 import 有關的
- .edata 跟 export 有關
- .rsrc 跟 resource 有關
- .reloc 跟重定位有關
- .pdata 跟例外處理有關

IAT(Import Address Table)



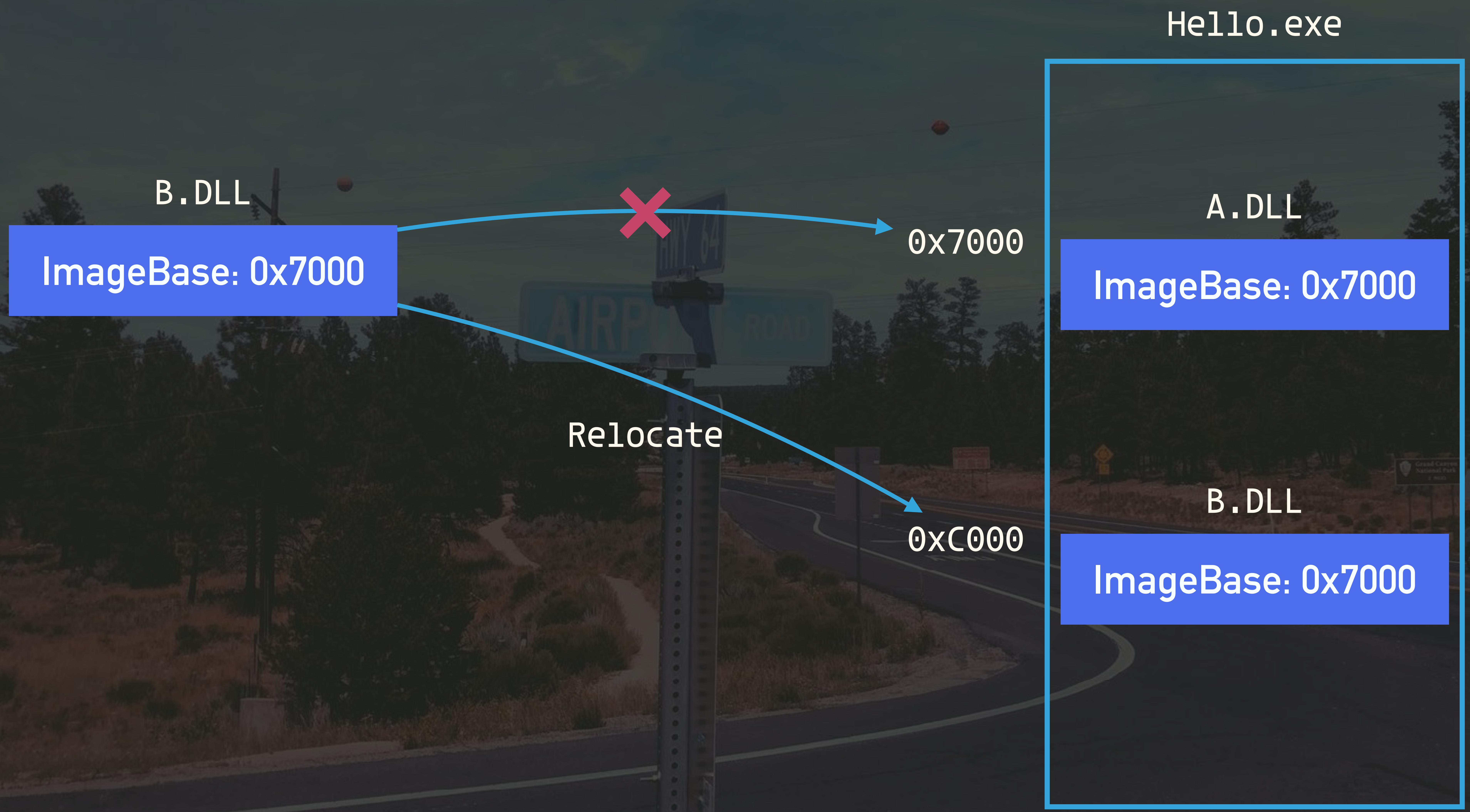
EAT(Export Address Table)



GetProcAddress() 如何使用 EAT 尋找 functions

- 先從 AddressOfNames 找到名字
- 使用第一步的 index 在 Ordinals 中找到對應的 ordinal 值
- 使用第二步的 ordinal 在 Funcitons 中尋找 function offset

Base Relocation Table



Base Relocation Table

- IMAGE_BASE_RELOCATION
- 由 VirtualAddress, SizeOfBlock, TypeOffset 構成
- TypeOffset, 16bit, high 4 bit for type, low 12 bit for offset
- VirtualAddress + offset 就是需要重定位的地方

.rscs

- Resource hacker

Run-Time Packer

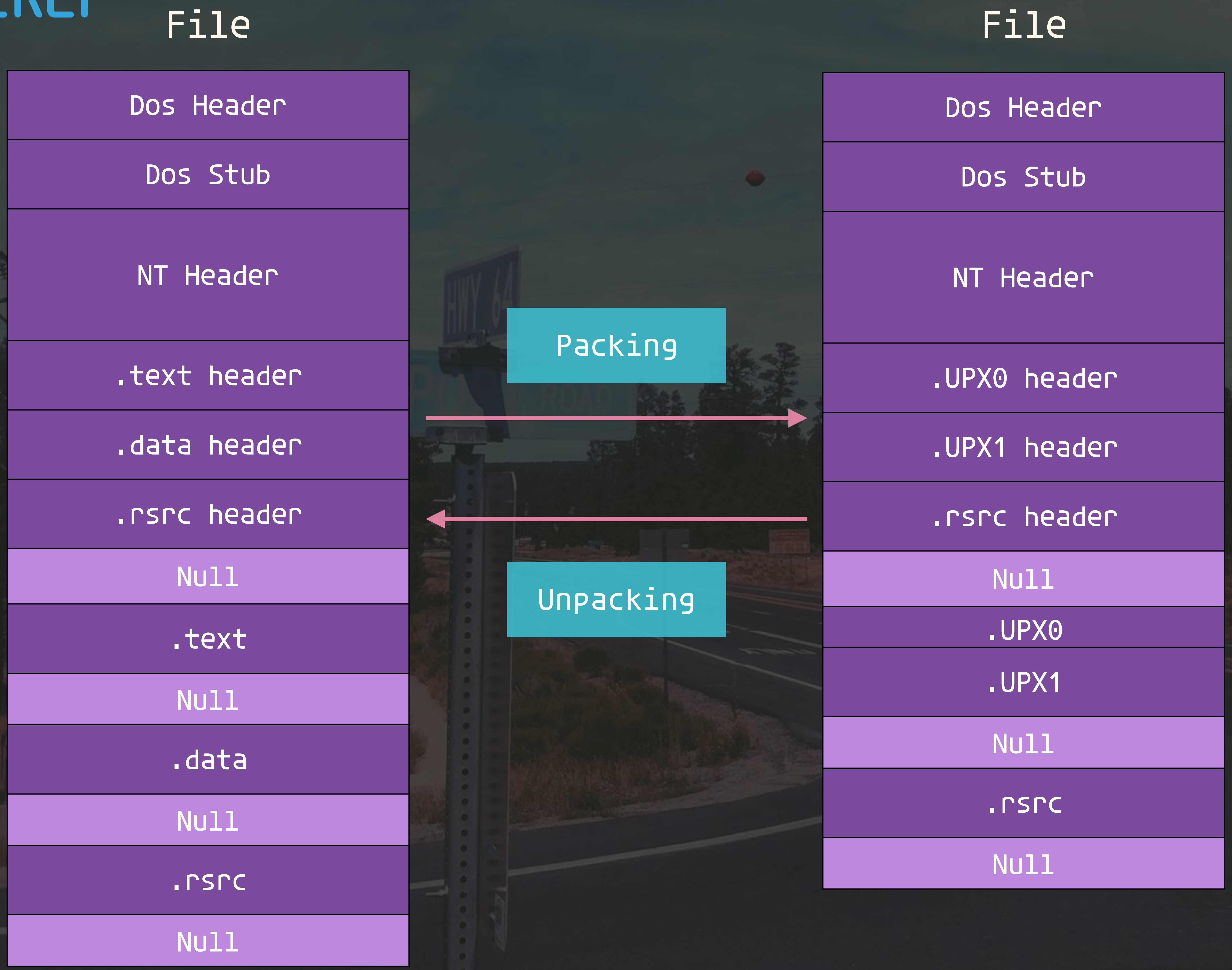
- To compress the executable

  e.g. UPX, ASPack

- To protect the executable

  e.g. VMProtect, ASProtect, Themida

Run-Time Packer



File

Dos Header

Dos Stub

NT Header

.text header

.data header

.rsrc header

Null

.text

Null

.data

Null

.rsrc

Null

Packing

Unpacking

File

Dos Header

Dos Stub

NT Header

.UPX0 header

.UPX1 header

.rsrc header

Null

.UPX0

.UPX1

Null

.rsrc

Null