v1

Engenharia reversa para principiantes

Conhecimentos e habilidades necessárias para iniciar-se na arte da escovação de bits

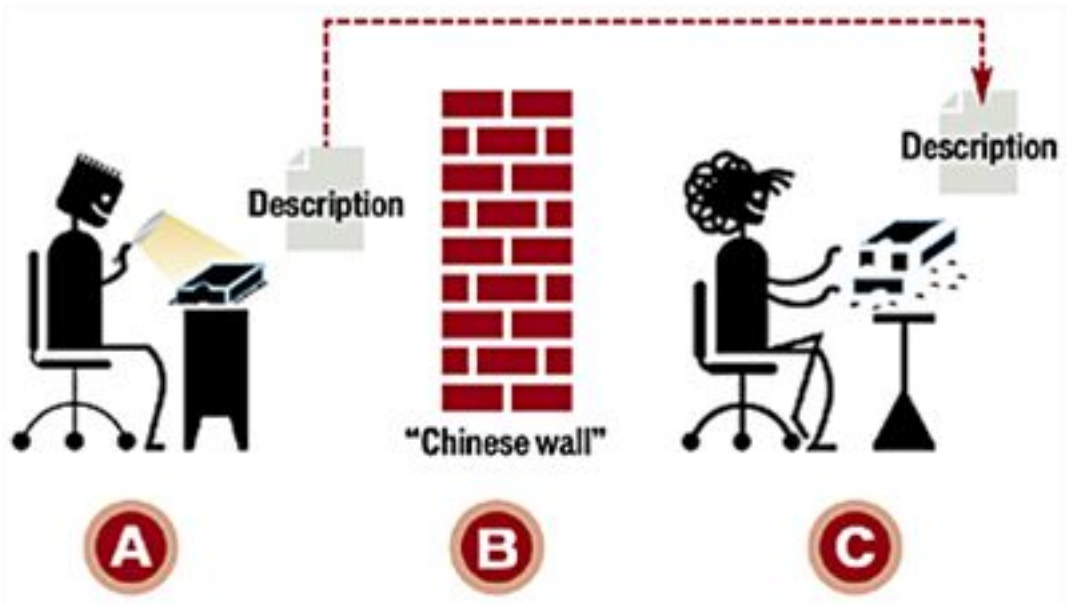
Wanderley Caloni

Objetivos

- Iniciante
  - Apontar caminhos de aprendizado
- Intermediário
  - Preencher lacunas de conhecimento
  - Gerar dúvidas e questões para serem respondidas

O que é engenharia reversa

Descobrir os princípios tecnológicos de um dispositivo, um objeto ou um sistema (programa).



Sem palavras bonitas: descobrir como funciona.

Onde a engenharia reversa é usada

- Empresas de antivírus (vacina)
- Análise de vulnerabilidades (*service-packs*)
- *Debugging* (e.g. blogue do Mark Russinovich)
- Formatos proprietários (e.g. OpenOffice)
- SOs fechados (e.g. projeto Wine)

Conhecimentos necessários

- Programação
  - Conceitos básicos
  - APIs
  - Arquitetura

Conhecimentos necessários

- Sistema operacional
  - Sistema de janelas
  - Processos e *threads*
  - Gerenciamento de memória

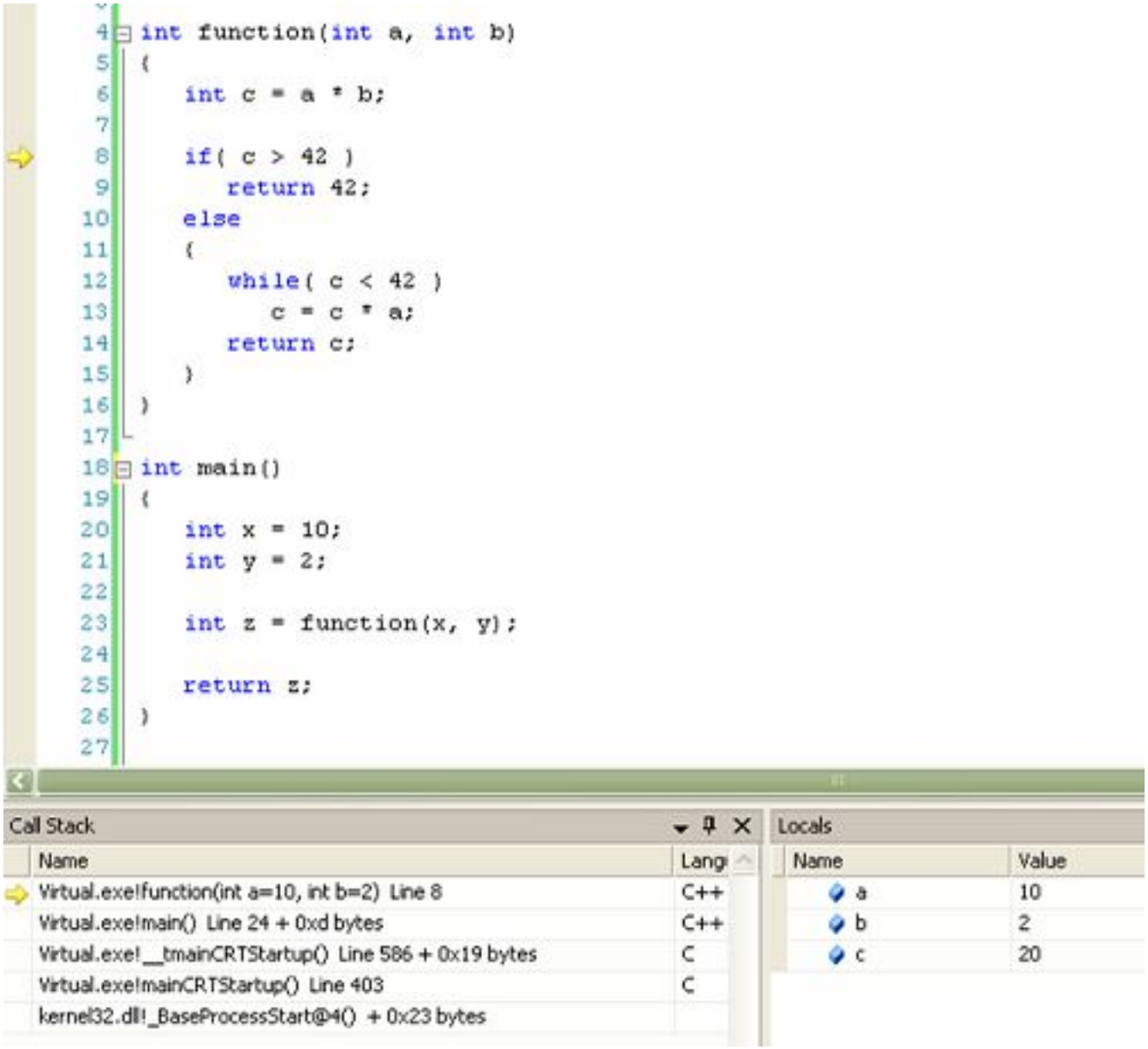
Conhecimentos necessários

- Ferramentas
  - Depuradores
  - *Disassemblers*
  - *Loggers*

Programação

- Conceitos básicos
- APIs
- Arquitetura

Conceitos básicos de programação



```
4  int function(int a, int b)
5  {
6      int c = a * b;
7
8      if( c > 42 )
9          return 42;
10     else
11     {
12         while( c < 42 )
13             c = c * a;
14         return c;
15     }
16 }
17
18 int main()
19 {
20     int x = 10;
21     int y = 2;
22
23     int z = function(x, y);
24
25     return z;
26 }
27
```

Call Stack

Name	Lang
Virtual.exe!function(int a=10, int b=2) Line 8	C++
Virtual.exe!main() Line 24 + 0xd bytes	C++
Virtual.exe!__tmainCRTStartup() Line 586 + 0x19 bytes	C
Virtual.exe!mainCRTStartup() Line 403	C
kernel32.dll!_BaseProcessStart@4() + 0x23 bytes	

Locals

Name	Value
a	10
b	2
c	20

Conceitos básicos de programação

- Aprenda lógica
- Aprenda C
  - C Completo e Total (Schildt)
  - Treinamento em linguagem C (Mizrahi)
  - C - A Linguagem de Programação (R&K)
- Programe
  - Programe
    - Programe

APIs

GetWindowText Function

The **GetWindowText** function copies the text of the specified window's title bar (if it has one) into a buffer. If the specified window is a control, the text of the control is copied. However, **GetWindowText** cannot retrieve the text of a control in another application.

Syntax

```
int GetWindowText(
    HWND hWnd,
    LPTSTR lpString,
    int nMaxCount
);
```

Parameters

hWnd
[in] Handle to the window or control containing the text.

lpString
[out] Pointer to the buffer that will receive the text. If the string is as long or longer than the buffer, the string is truncated and terminated with a NULL character.

nMaxCount
[in] Specifies the maximum number of characters to copy to the buffer, including the NULL character. If the text exceeds this limit, it is truncated.

Return Value

If the function succeeds, the return value is the length, in characters, of the copied string, not including the terminating NULL character. If the window has no title bar or text, if the title bar is

APIs

- Aprenda a programar para seu SO preferido
  - Programming Windows (Petzold)
  - Advanced Programming In The UNIX Environment (Stevens)
- Use a documentação como referência
  - MSDN
- Programe
  - Programe
    - Programe

Arquitetura

```
int function(int a, int b)
{
00411240  push        ebp
00411241  mov         ebp,esp
00411243  sub         esp,44h
00411246  push        ebx
00411247  push        esi
00411248  push        edi
   int c = a * b;
00411249  mov         eax,dword ptr [a]
0041124C  imul        eax,dword ptr [b]
00411250  mov         dword ptr [c],eax

   if( c > 42 )
00411253  cmp         dword ptr [c],2Ah
00411257  jle         function+22h (411262h)
      return 42;
00411259  mov         eax,2Ah
0041125E  jmp         function+37h (411277h)
   else
00411260  jmp         function+37h (411277h)
   {
      while( c < 42 )
00411262  cmp         dword ptr [c],2Ah
00411266  jge         function+34h (411274h)
         c = c * a;
00411268  mov         eax,dword ptr [c]
0041126B  imul        eax,dword ptr [a]
0041126F  mov         dword ptr [c],eax
```

Arquitetura

- Aprenda os conceitos básicos da arquitetura preferida
  - Introdução a microprocessadores
  - Guia do Programador para IBM PC (Norton)
- Memória e pilha
- *Assembly*
  - The Art of Assembly Language (Hyde)
- Faça programas simples em *assembly*

Sistema operacional

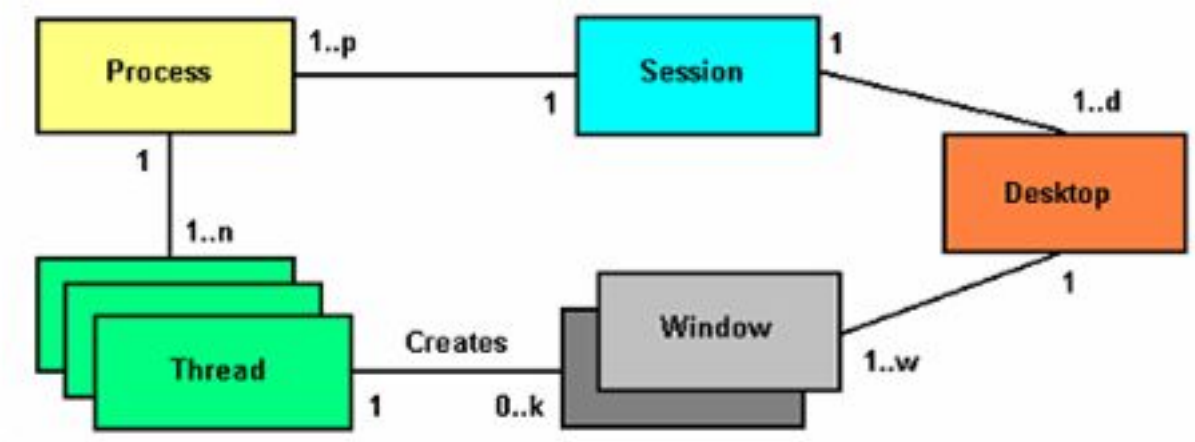
- Arquitetura
- Sistema de janelas
- Processos e *threads*
- Gerenciamento de memória

Arquitetura do sistema operacional

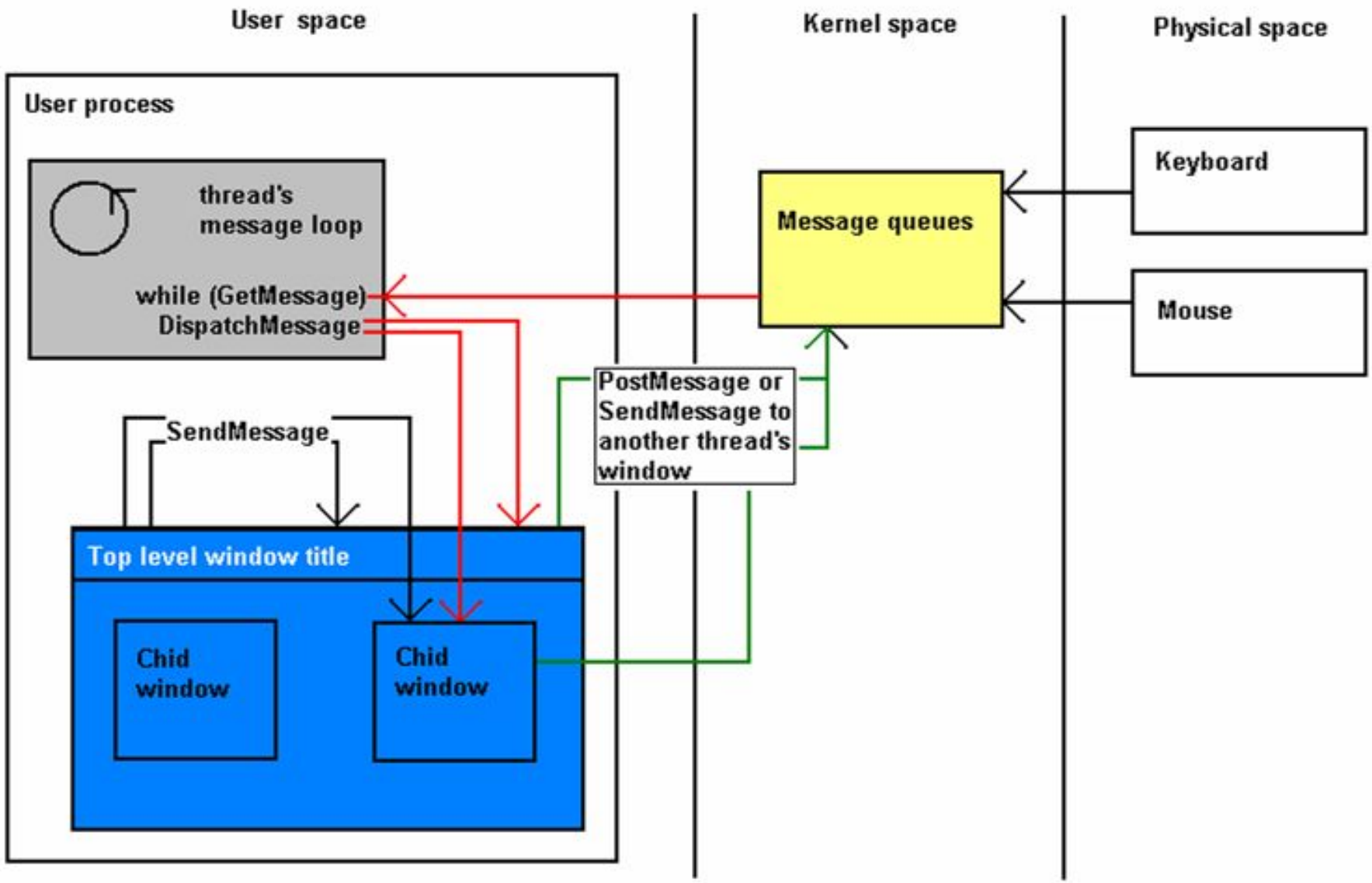
- Divisão modular do sistema operacional
  - Microsoft Windows Internals (Russinovich)
- Testes no funcionamento interno do SO
  - Ferramentas da Sysinternals (sysinternals.com)
- Funcionamento de *drivers*
  - DriverEntry.com.br

Sistema de janelas

Cada sessão possui **d** *desktops* e cada *desktop* possui **w** janelas. Cada sessão possui **p** processos, que possuem **n** *threads*, que controlam **w** janelas.



Sistema de janelas



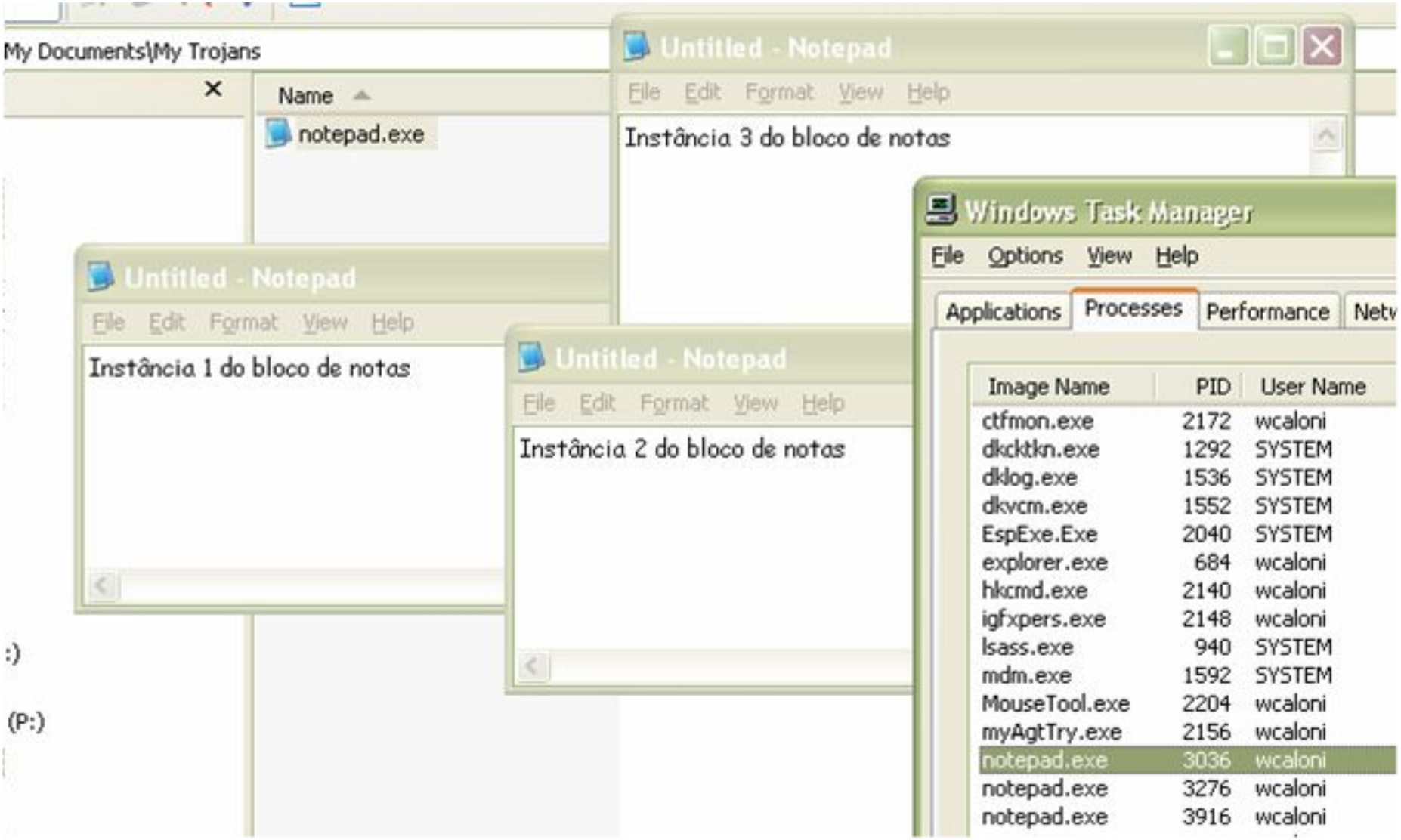
Sistema de janelas

- Programar para Windows
  - Programming Windows (Petzold)
- Testar relação e mensagens entre janelas e *threads*
  - Ferramenta Spy++ (Microsoft Visual Studio)

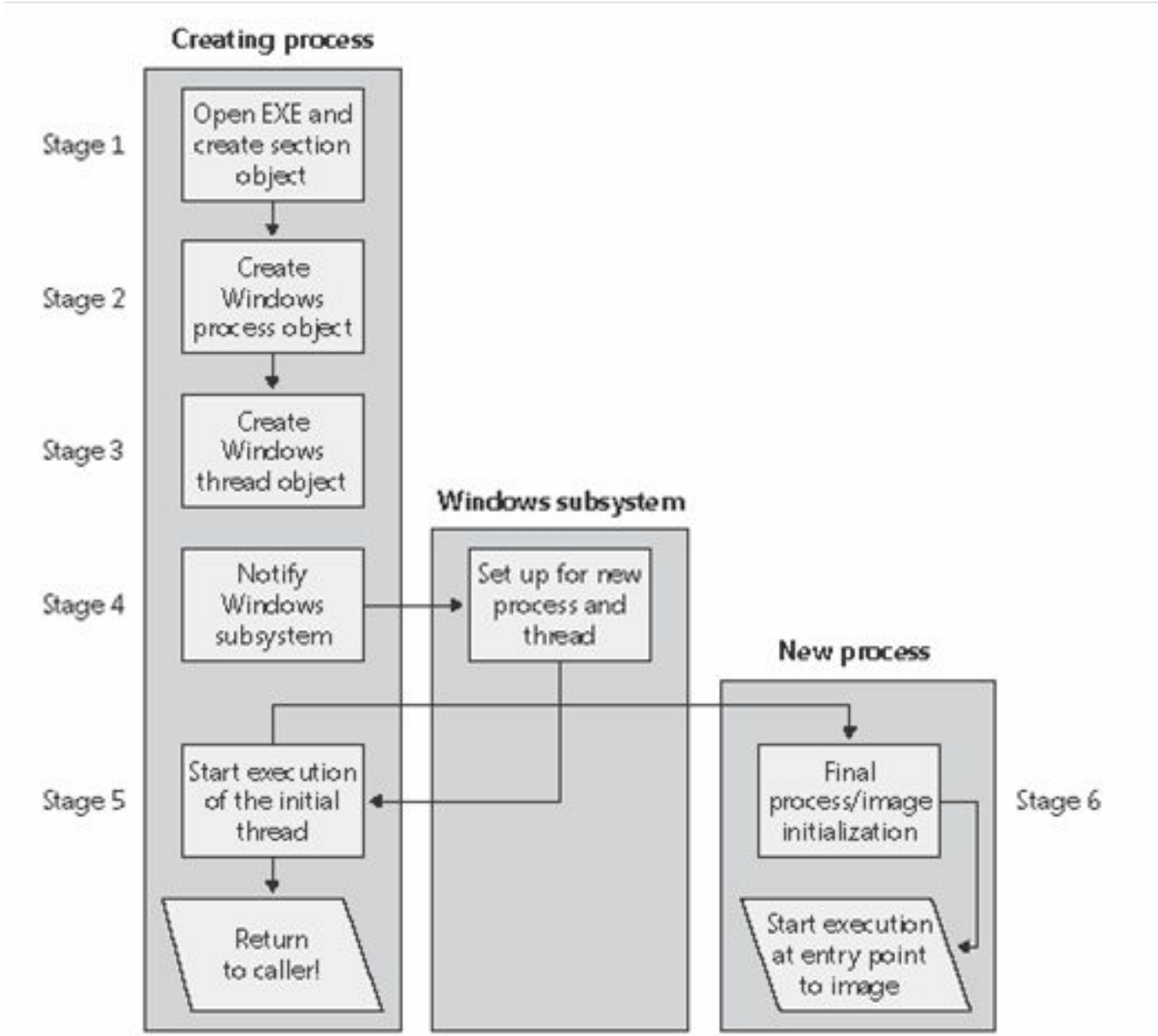
Processos e *threads*

- Um processo é um espaço de memória no sistema que contém uma ou mais linhas de execução (*thread*).
- Uma *thread* é uma linha de execução que roda no contexto de um determinado processo.

Processos de um mesmo executável



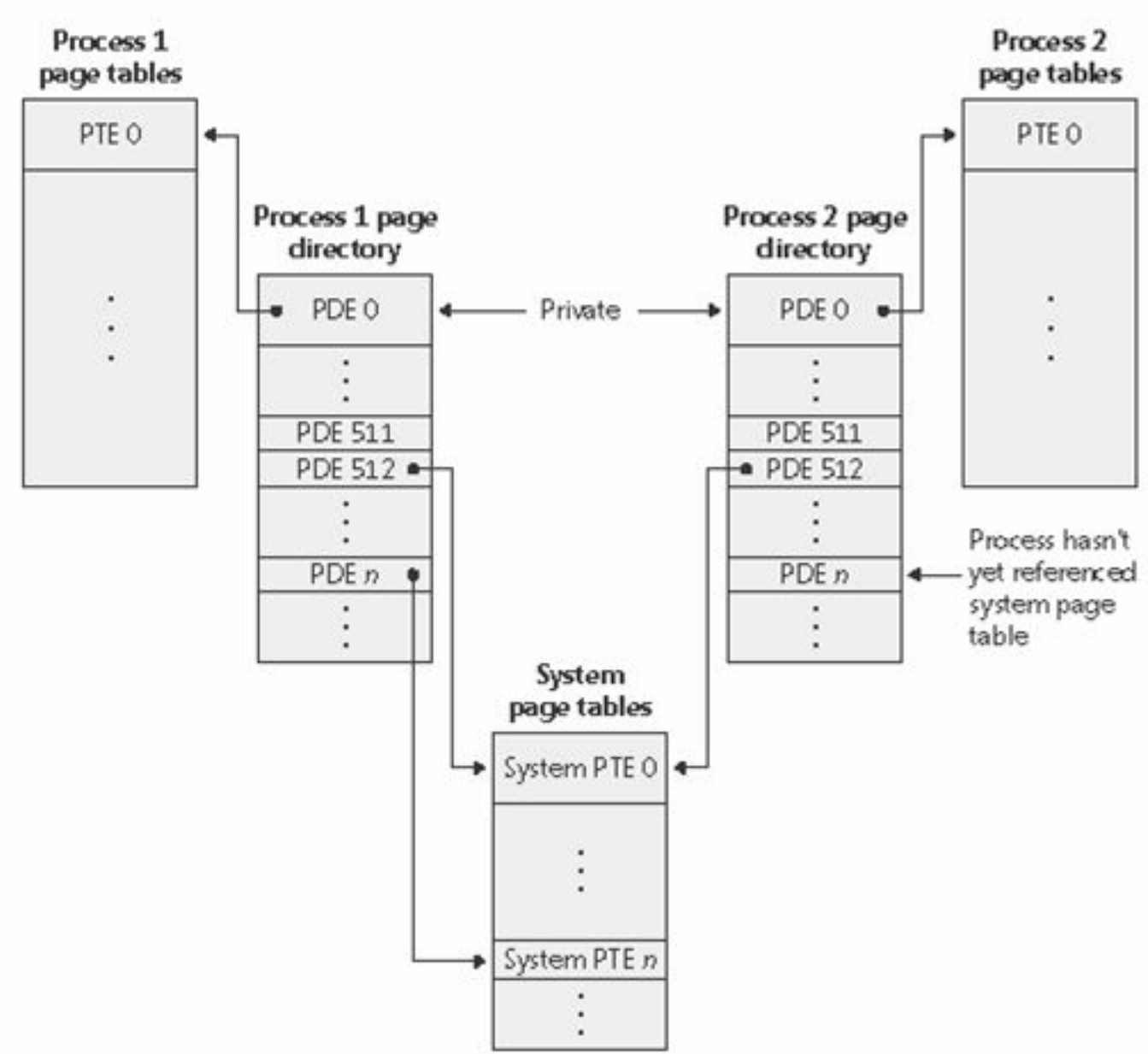
Criação de um processo



Gerenciamento de memória

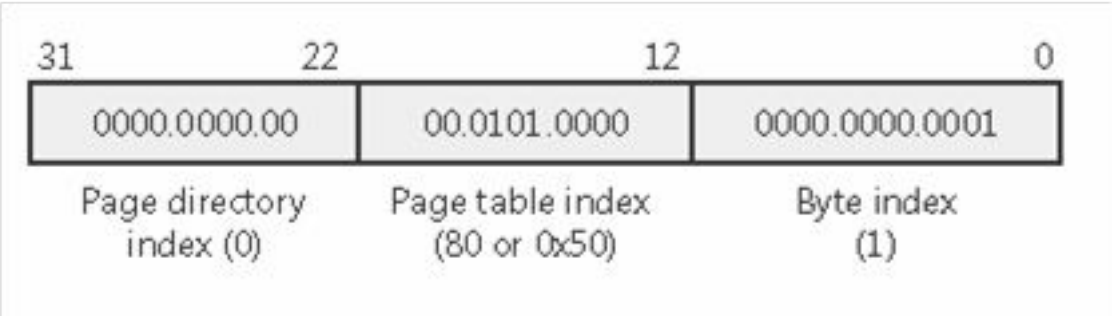
- É função do sistema operacional gerenciar a memória física e torná-la disponível para as aplicações.
- A visão de uma aplicação quanto à memória se trata de uma abstração chamada de memória virtual.

Gerenciamento de memória

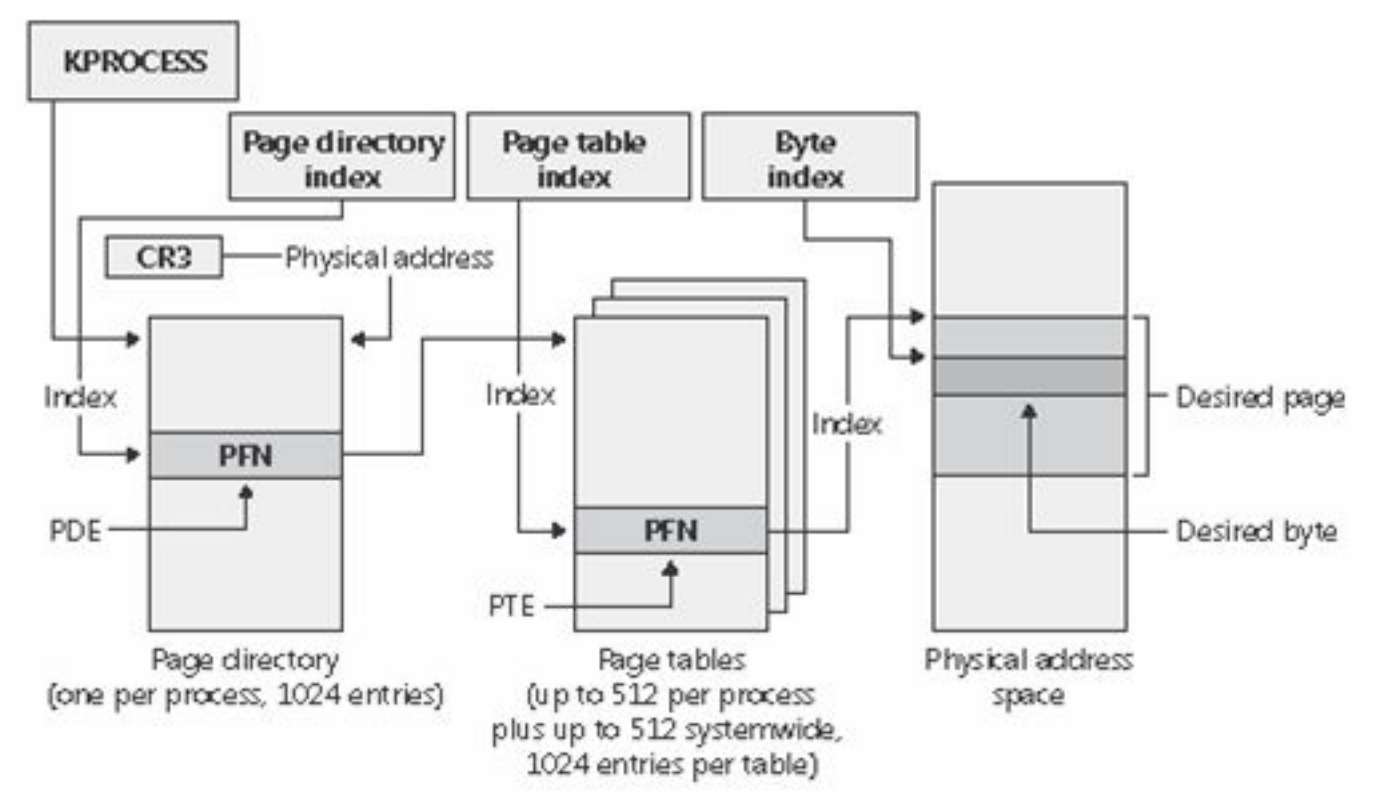


Gerenciamento de memória

Um endereço virtual é um índice que se traduz em um bloco de memória física alocada.



Gerenciamento de memória



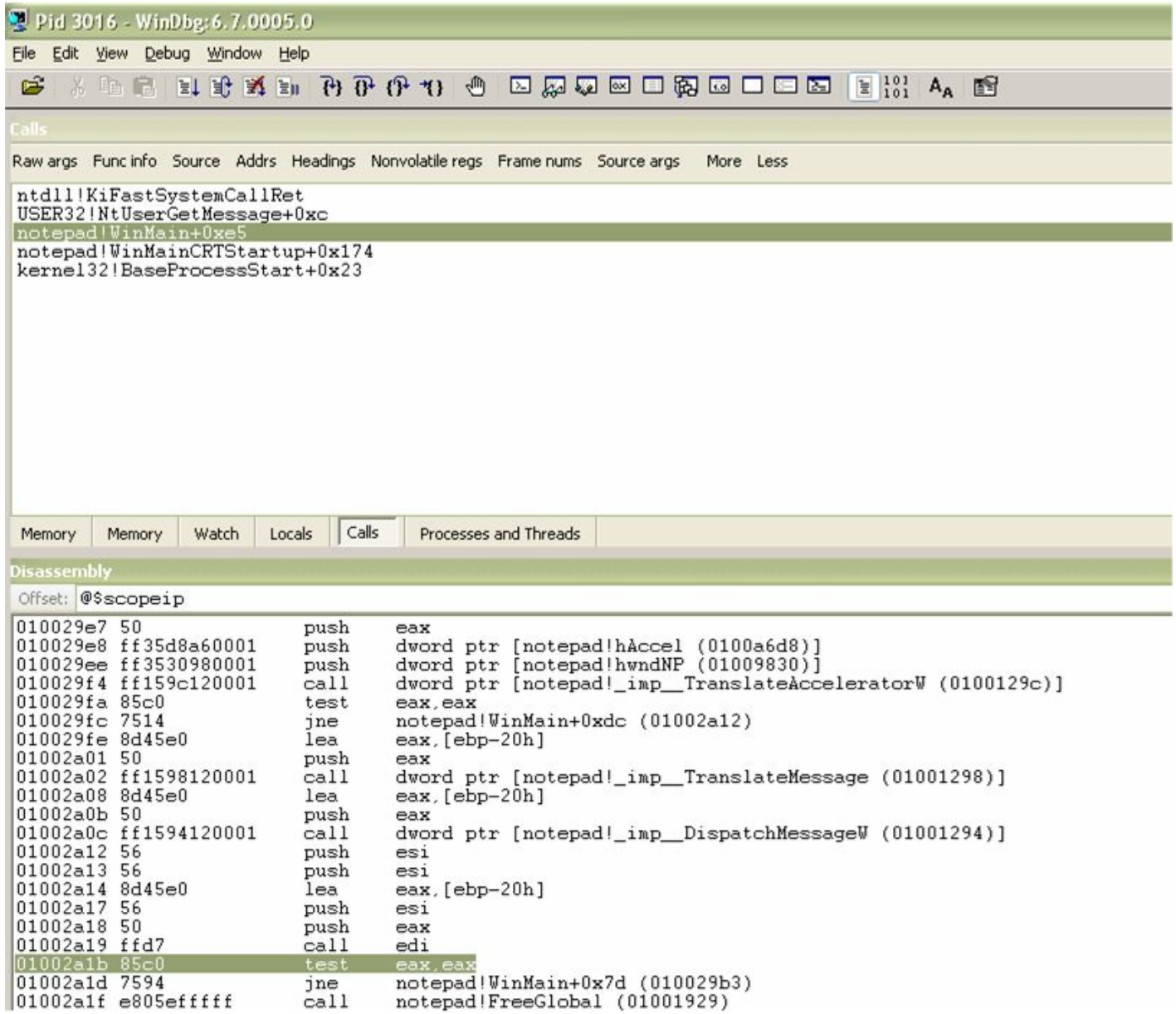
Gerenciamento de memória

- Mais informações:
  - 1bit.com.br (palestra "Por dentro do Windows: Gerenciamento de Memória")

Ferramentas

- Depuradores
  - WinDbg
  - OllyDbg
- *Disassemblers*
  - IDA
  - OllyDbg
- *Loggers*
  - File Monitor
  - Registry Monitor
  - Process Monitor
- Adicionais
  - HxD
  - Process Explorer

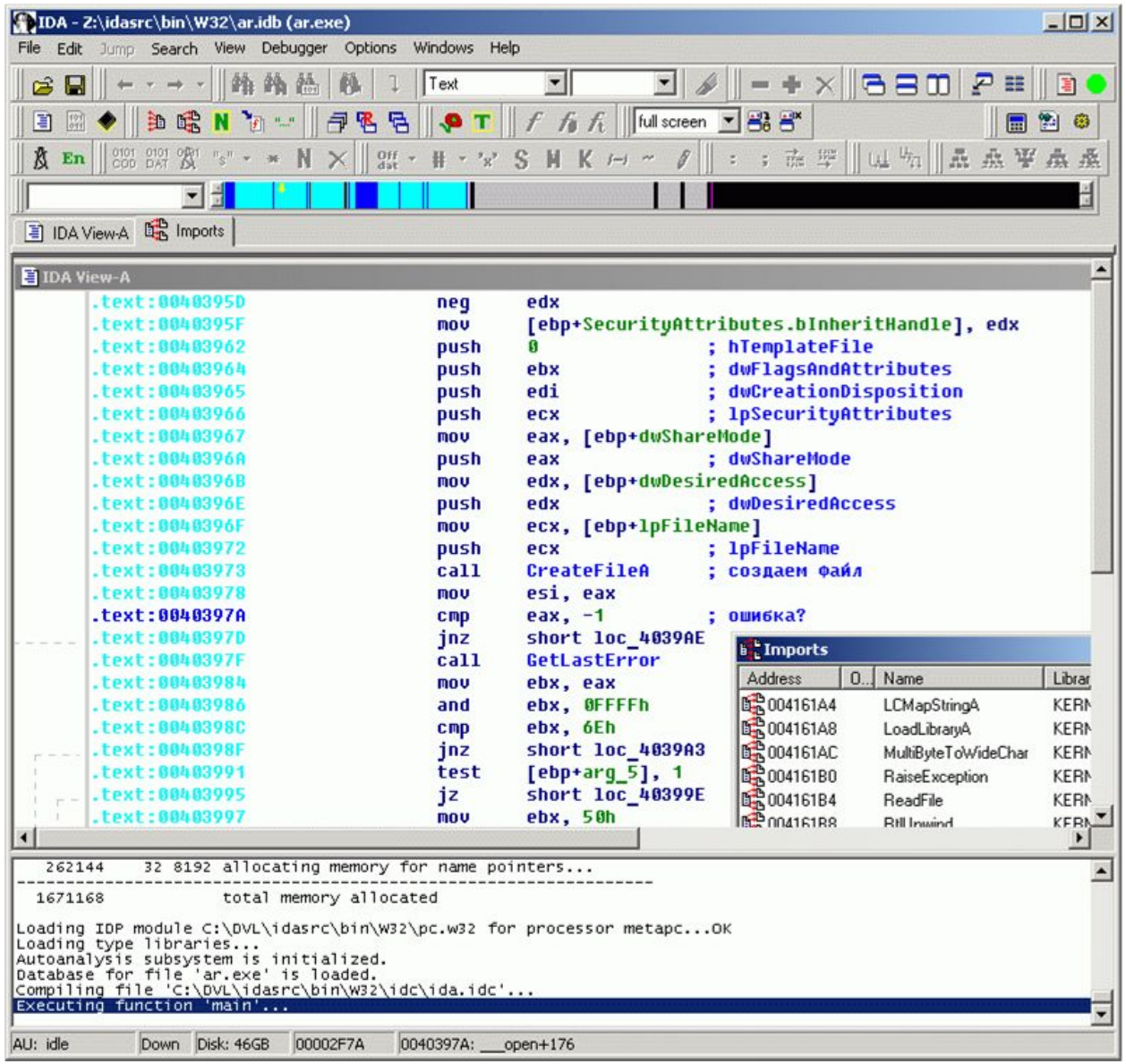
WinDbg


Pid 3016 - WinDbg:6.7.0005.0
File Edit View Debug Window Help
Calls
Raw args Func info Source Addrs Headings Nonvolatile regs Frame nums Source args More Less
ntdll!KiFastSystemCallRet
USER32!NtUserGetMessage+0xc
notepad!WinMain+0xe5
notepad!WinMainCRTStartup+0x174
kernel32!BaseProcessStart+0x23
Memory Memory Watch Locals Calls Processes and Threads
Disassembly
Offset: @$scopeip
010029e7 50 push eax
010029e8 ff35d8a60001 push dword ptr [notepad!hAccel (0100a6d8)]
010029ee ff3530980001 push dword ptr [notepad!hwndNP (01009830)]
010029f4 ff159c120001 call dword ptr [notepad!_imp__TranslateAcceleratorW (0100129c)]
010029fa 85c0 test eax,eax
010029fc 7514 jne notepad!WinMain+0xdc (01002a12)
010029fe 8d45e0 lea eax,[ebp-20h]
01002a01 50 push eax
01002a02 ff1598120001 call dword ptr [notepad!_imp__TranslateMessage (01001298)]
01002a08 8d45e0 lea eax,[ebp-20h]
01002a0b 50 push eax
01002a0c ff1594120001 call dword ptr [notepad!_imp__DispatchMessageW (01001294)]
01002a12 56 push esi
01002a13 56 push esi
01002a14 8d45e0 lea eax,[ebp-20h]
01002a17 56 push esi
01002a18 50 push eax
01002a19 ffd7 call edi
01002a1b 85c0 test eax,eax
01002a1d 7594 jne notepad!WinMain+0x7d (010029b3)
01002a1f e805efffff call notepad!FreeGlobal (01001929)

WinDbg

- Vem com o pacote "Debugging Tools"
- Interface amigável para depuradores console
- Depura *kernel mode* (núcleo do SO)
- Extensões poderosas

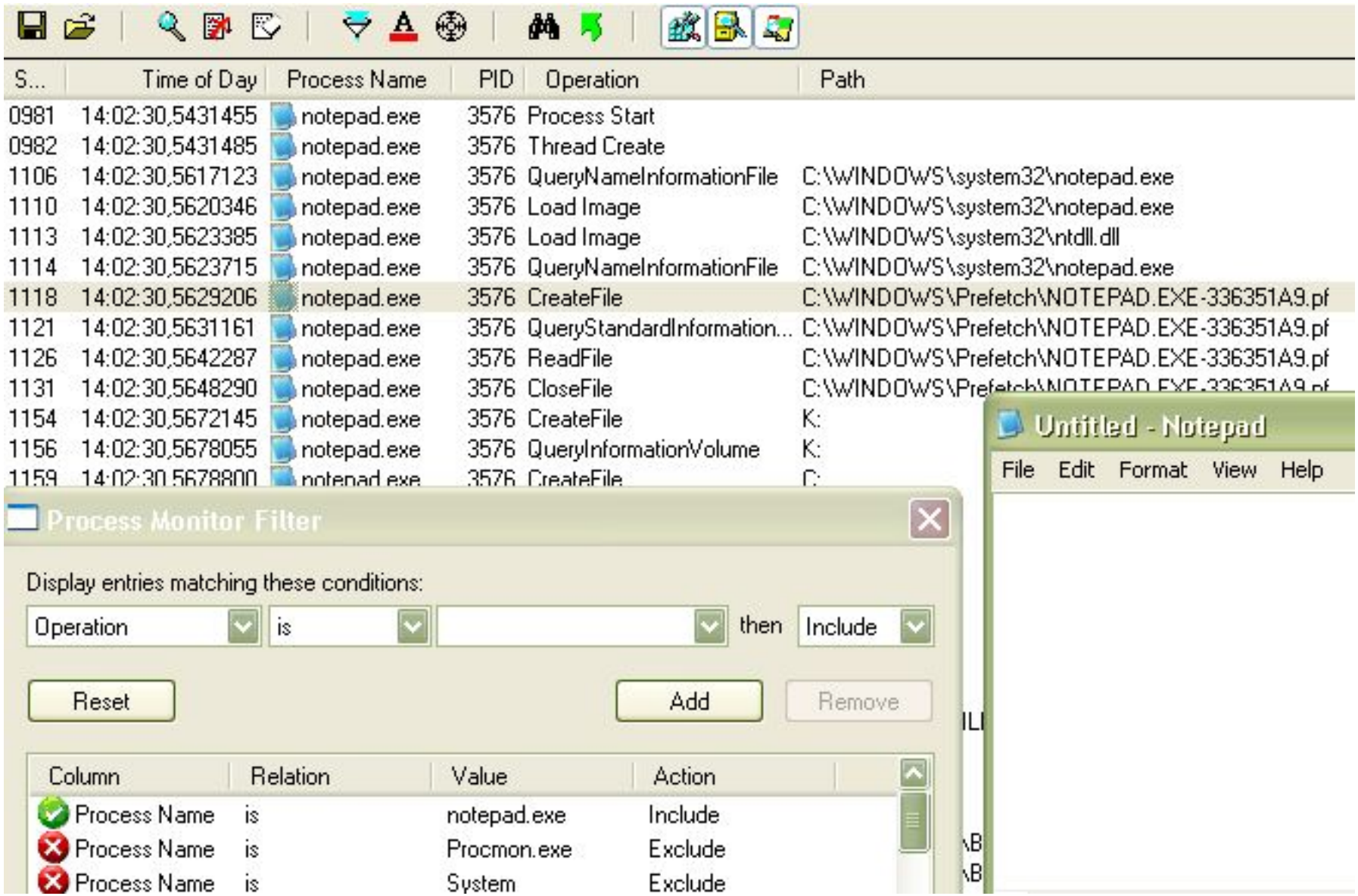
IDA


IDA - Z:\idasrc\bin\W32\ar.idb (ar.exe)
File Edit Jump Search View Debugger Options Windows Help
IDA View-A
Imports
.text:0040395D neg edx
.text:0040395F mov [ebp+SecurityAttributes.bInheritHandle], edx
.text:00403962 push 0 ; hTemplateFile
.text:00403964 push ebx ; dwFlagsAndAttributes
.text:00403965 push edi ; dwCreationDisposition
.text:00403966 push ecx ; lpSecurityAttributes
.text:00403967 mov eax, [ebp+dwShareMode]
.text:0040396A push eax ; dwShareMode
.text:0040396B mov edx, [ebp+dwDesiredAccess]
.text:0040396E push edx ; dwDesiredAccess
.text:0040396F mov ecx, [ebp+lpFileName]
.text:00403972 push ecx ; lpFileName
.text:00403973 call CreateFileA ; создаем файл
.text:00403978 mov esi, eax
.text:0040397A cmp eax, -1 ; ошибка?
.text:0040397D jnz short loc_4039AE
.text:0040397F call GetLastError
.text:00403984 mov ebx, eax
.text:00403986 and ebx, 0FFFFh
.text:0040398C cmp ebx, 6Eh
.text:0040398F jnz short loc_4039A3
.text:00403991 test [ebp+arg_5], 1
.text:00403995 jz short loc_40399E
.text:00403997 mov ebx, 50h
Imports
Address O... Name Librar
004161A4 LCMapStringA KERN
004161A8 LoadLibraryA KERN
004161AC MultiByteToWideChar KERN
004161B0 RaiseException KERN
004161B4 ReadFile KERN
262144 32 8192 allocating memory for name pointers...
1671168 total memory allocated
Loading IDP module C:\DVL\idasrc\bin\W32\pc.w32 for processor metapc...OK
Loading type libraries...
Autoanalysis subsystem is initialized.
Database for file 'ar.exe' is loaded.
Compiling file 'C:\DVL\idasrc\bin\W32\idc\ida.idc'...
Executing function 'main'...
AU: idle Down Disk: 46GB 00002F7A 0040397A: __open+176

IDA

- Disassembler estático e dinâmico
- Analisa código de acordo com chamadas da API
- Facilita reconstrução de código
- Cria patches (remendos para o executável original)
- Gera gráfico de chamadas

Process Monitor



Process Monitor

- Monitora registro, arquivos e processos
- Possui filtros extremamente poderosos
- Consegue monitorar *boot* do SO

Tarefas de aprimoramento

- Resolver problemas em seu sistema operacional
  - Travamento de programas
  - Lentidão não esperada
  - *Crash* do SO
- Quebrar proteção de programas (para aprendizado)
  - *Antidebugging*
  - Telas com senha
  - Licenciamento
- Entender funcionamento de vírus
  - Fontes de vírus conhecidos
  - Ataques recebidos por *email*
  - Desenvolvimento de ataques (para aprendizado)
- Faça você mesmo (não espere por ajuda)

Dúvidas?

http://www.caloni.com.br/blog

2007-10-10