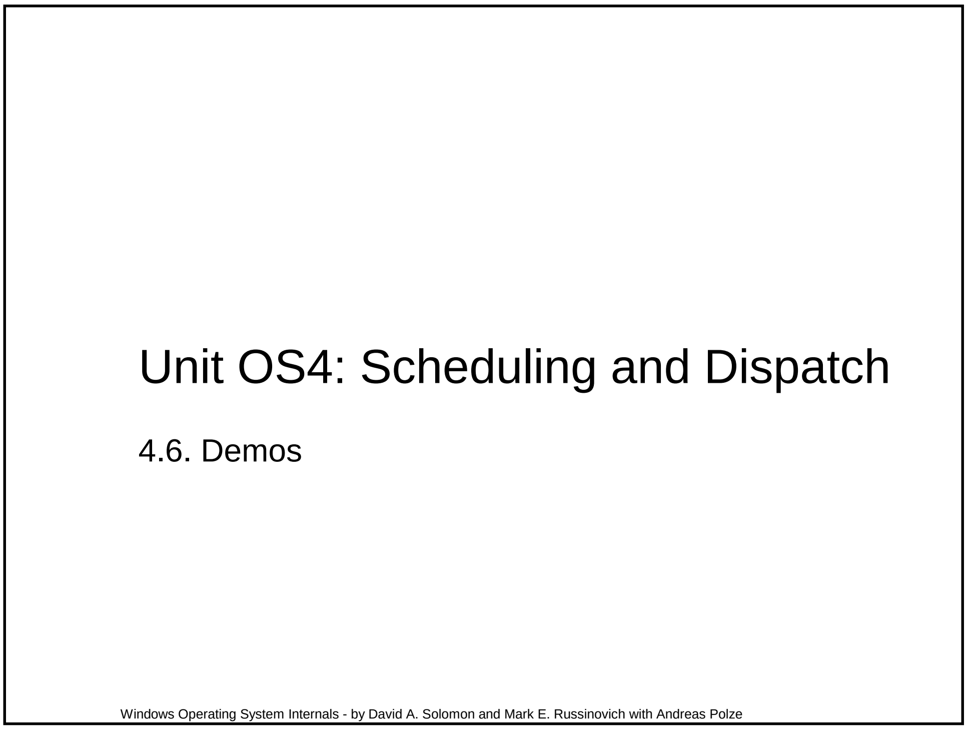
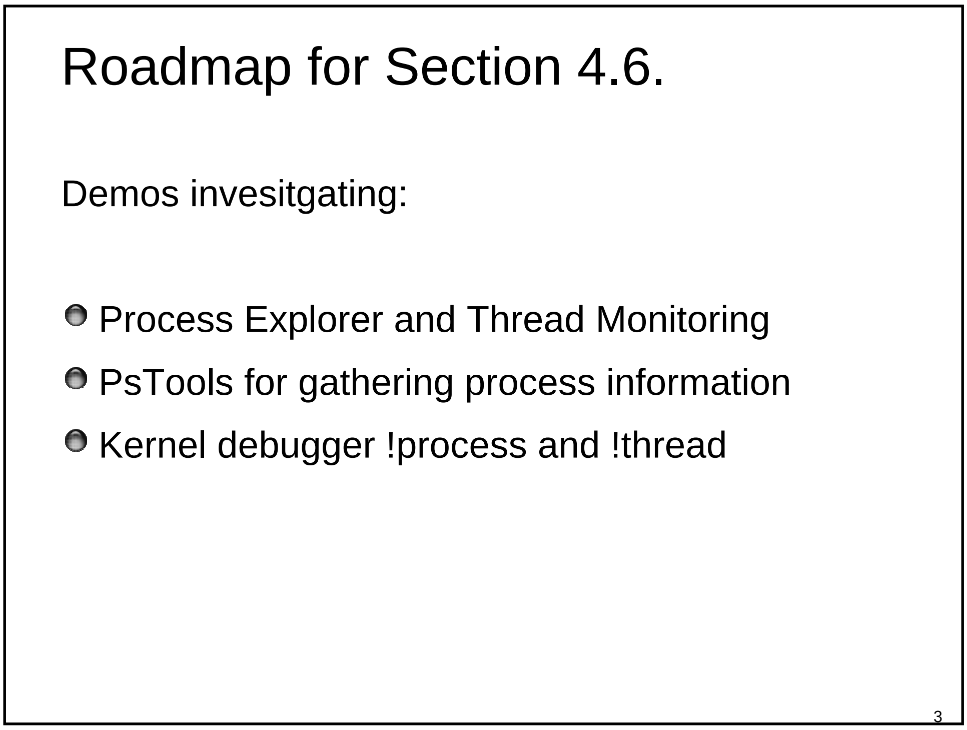


Unit OS4: Scheduling and Dispatch

4.6. Demos

Windows Operating System Internals - by David A. Solomon and Mark E. Russinovich with Andreas Polze



Roadmap for Section 4.6.

Demos invesitgating:

- Process Explorer and Thread Monitoring
- PsTools for gathering process information
- Kernel debugger !process and !thread

3

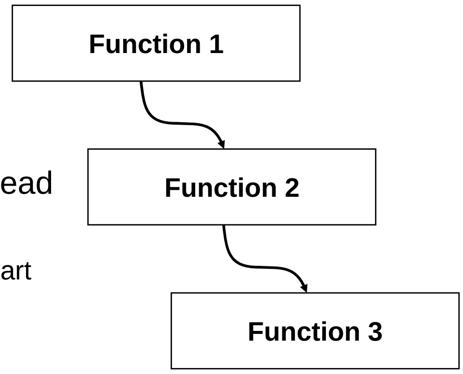
Lab: Refresh Highlighting

1. Change update speed to paused by pressing space bar
2. Run Notepad
3. In ProcExp, hit F5 and notice new process
4. Exit Notepad
5. In ProcExp, hit F5 and notice Notepad in red

- Uses
  - Understanding process startup sequences
  - Detecting appearance of processes coming and going

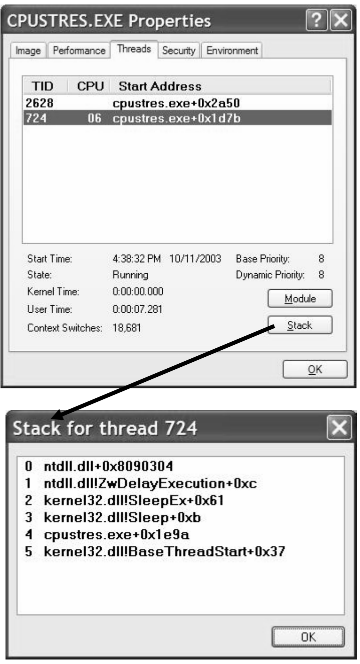
Call Stacks

- Process Explorer can also show the thread call stack
  - Represents sequence of functions called
- Important if start address doesn't indicate what the thread is doing
  - E.g. if it's a generic library start routine



Call Stacks

- Click Stack to view call stack
  - Lists functions in reverse chronological order
- Note that start address on Threads tab is different than first function shown in stack
  - This is because all user threads start in a Windows library function which calls the programmed start address



CPUSTRES.EXE Properties

TID	CPU	Start Address
2628		cpustres.exe+0x2a50
724	06	cpustres.exe+0x1d7b

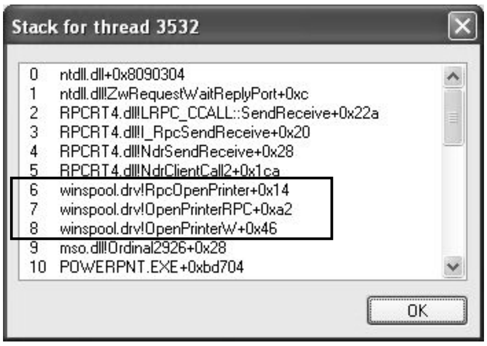
Start Time: 4:38:32 PM 10/11/2003
Base Priority: 8
State: Running
Dynamic Priority: 8
Kernel Time: 0:00:00.000
User Time: 0:00:07.281
Context Switches: 18,681

Stack for thread 724

```
0  ntdll.dll+0x8090304
1  ntdll.dll!ZwDelayExecution+0xc
2  kernel32.dll!SleepEx+0x61
3  kernel32.dll!Sleep+0xb
4  cpustres.exe+0x1e9a
5  kernel32.dll!BaseThreadStart+0x37
```

Example: Viewing Stacks

- Problem: Powerpoint was hanging for 1 minute on startup
- Thread stack shows waiting on a printer driver



Stack for thread 3532

```
0   ntdll.dll+0x8090304
1   ntdll.dll!ZwRequestWaitReplyPort+0xc
2   RPCRT4.dll!LRPC_CCALL::SendReceive+0x22a
3   RPCRT4.dll!I_RpcSendReceive+0x20
4   RPCRT4.dll!NdrSendReceive+0x28
5   RPCRT4.dll!NdrClientCall2+0x1ca
6   winspool.drv!RpcOpenPrinter+0x14
7   winspool.drv!OpenPrinterRPC+0xa2
8   winspool.drv!OpenPrinterW+0x46
9   mso.dll!Ordinal2926+0x28
10  POWERPNT.EXE+0xbd704
```

Suspending Processes

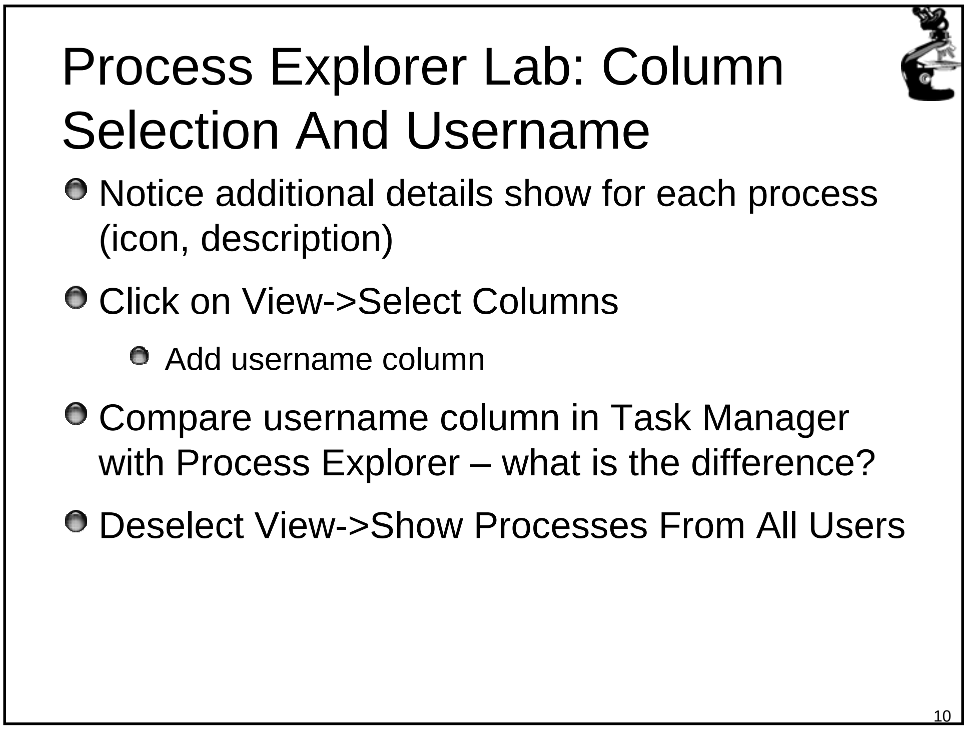
- Process Explorer can suspend a process
- Why would you want to do this?
  - You’ve started a long running job but want to pause it to do something else
    - Lowering the priority still leaves it running…
  - You’ve started a long download but want to have your network bandwidth temporarily
  - Some multi-service system process activity is due to other processes calling upon their services
    - Suspend a process that is consuming CPU time to see what that does to the system process in question

8

Lab: Suspend

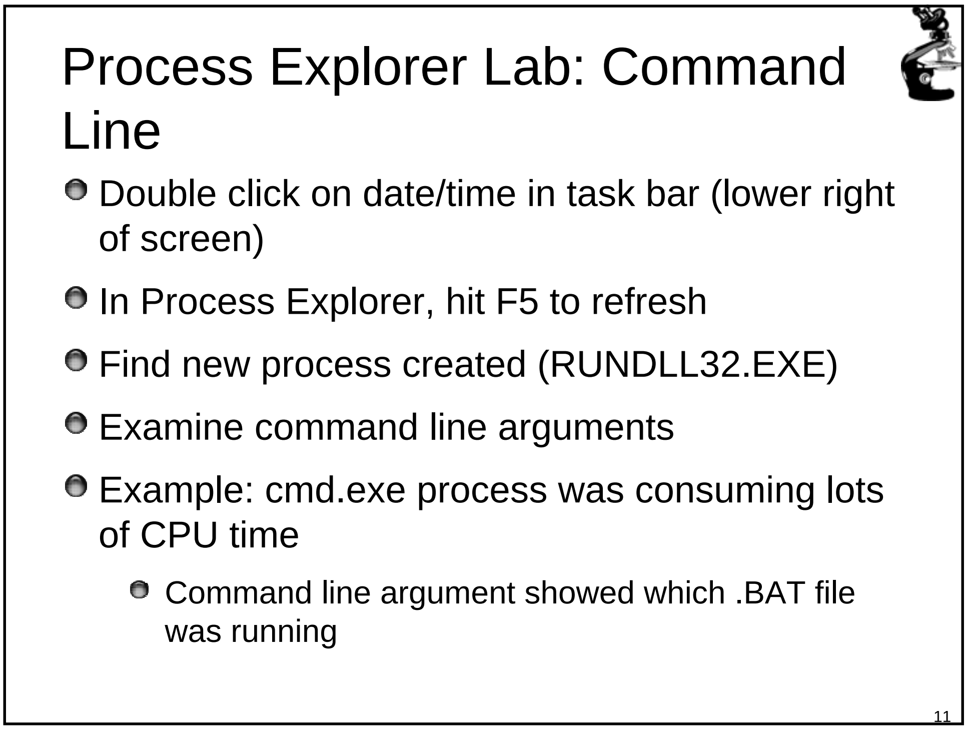
- Start Notepad
- From a command prompt:
  1. Suspend Notepad process with Process Explorer
  2. Try to switch back to Notepad (should not respond)
  3. Open Task Manager and look at Notepad’s status in the applications tab ☺
  4. Resume Notepad

9



Process Explorer Lab: Column Selection And Username

- Notice additional details show for each process (icon, description)
- Click on View->Select Columns
  - Add username column
- Compare username column in Task Manager with Process Explorer – what is the difference?
- Deselect View->Show Processes From All Users

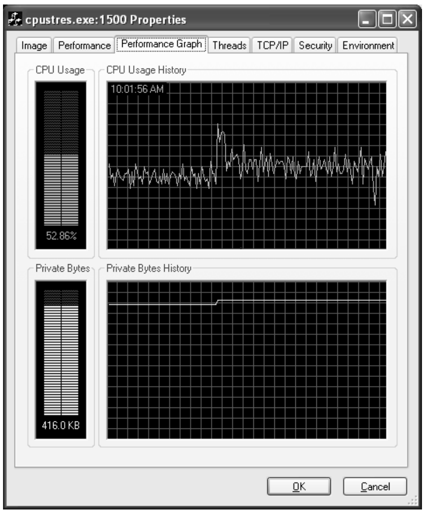


Process Explorer Lab: Command Line

- Double click on date/time in task bar (lower right of screen)
- In Process Explorer, hit F5 to refresh
- Find new process created (RUNDLL32.EXE)
- Examine command line arguments
- Example: cmd.exe process was consuming lots of CPU time
  - Command line argument showed which .BAT file was running

Examining CPU Time

- Open process properties and look at CPU usage history on the performance graph page
- Hover the mouse over a point to see the time of that value

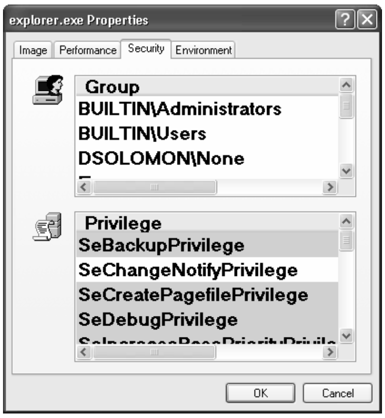


Process Explorer Lab: Environment Variables

- Open a command prompt
- Run Notepadexe from command prompt
- Type “set abc=xyz”
- In ProcExp, hit F5 and examine environment variables for Cmd.exe and Notepad.exe
  - Notice Notepad.exe does not know about the environment variable abc

Security

- Click on Security tab of process properties
- Shows rest of access token (username is on image tab)
  - Groups list
    - Includes OS-assigned groups
  - Privileges (user rights)
    - Disabled by default
    - Programs turn these on when needed
- This is really a “Resultant Set of Groups” and “Resultant Set of Privileges” page

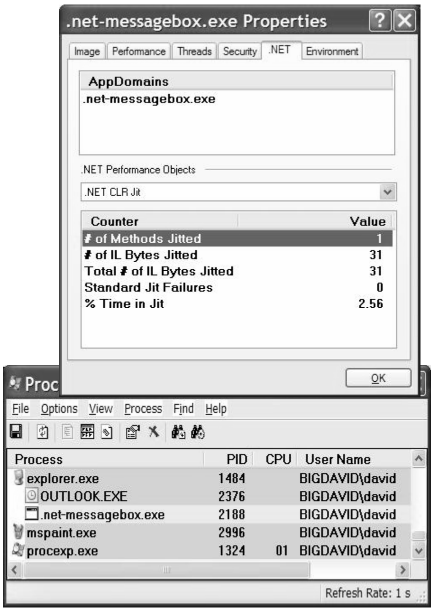


Group
BUILTIN\Administrators
BUILTIN\Users
DSOLOMON\None

Privilege
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreatePagefilePrivilege
SeDebugPrivilege
SeIncreaseBasePriorityPrivile

.NET Information

- Process Explorer is aware of .NET processes
  - Can highlight with Options->Highlight .NET Processes
- Process properties have .NET tab
  - Shows details about .NET process (CLR, Appdomains)
- Can also add .NET-specific columns to process list



AppDomains
.net-messagebox.exe

.NET Performance Objects: .NET CLR Jit

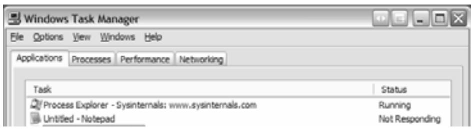
Counter	Value
# of Methods Jitted	1
# of IL Bytes Jitted	31
Total # of IL Bytes Jitted	31
Standard Jit Failures	0
% Time in Jit	2.56

Process	PID	CPU	User Name
explorer.exe	1484		BIGDAVID\david
OUTLOOK.EXE	2376		BIGDAVID\david
.net-messagebox.exe	2188		BIGDAVID\david
mspaint.exe	2996		BIGDAVID\david
procexp.exe	1324	01	BIGDAVID\david

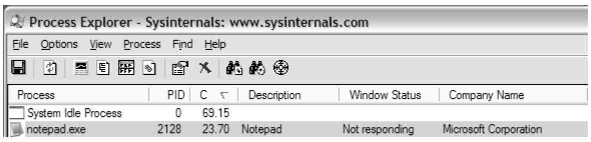
Refresh Rate: 1 s

Windows Status

- If you really like Task Manager's Applications tab:
  - Add the Window Title column
  - Add the Window Status column
    - Uses the same Windows function as Task Manager to determine status


Windows Task Manager
File Options View Windows Help
Applications Processes Performance Networking

Task	Status
Process Explorer - Sysinternals: www.sysinternals.com	Running
Untitled - Notepad	Not Responding


Process Explorer - Sysinternals: www.sysinternals.com
File Options View Process Find Help

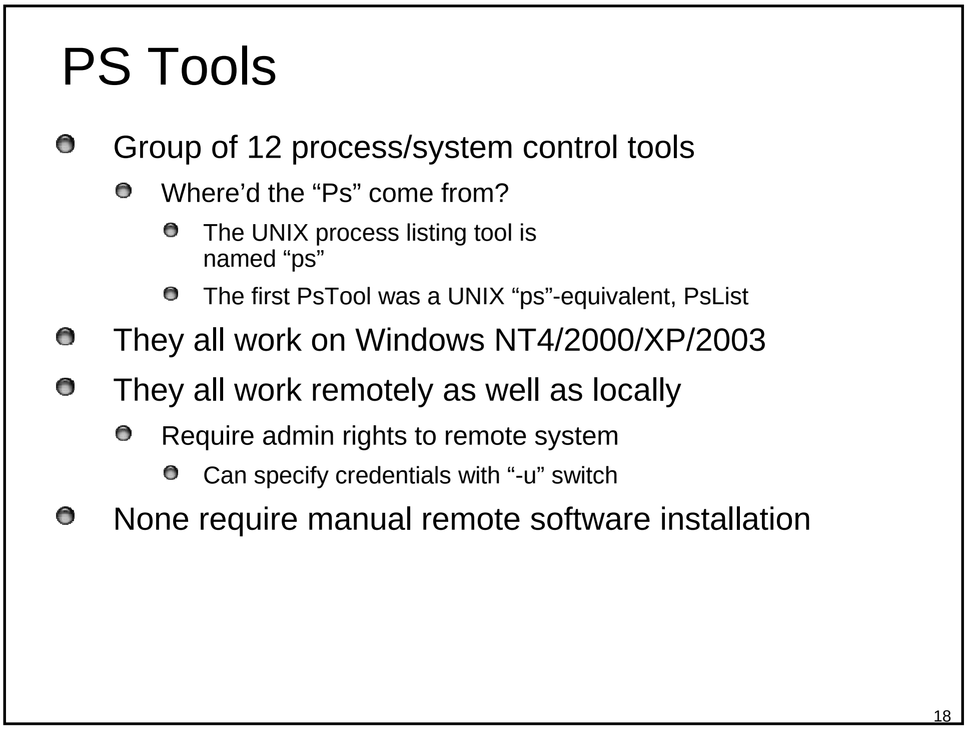
Process	PID	C	Description	Window Status	Company Name
System Idle Process	0	69.15			
notepad.exe	2128	23.70	Notepad	Not responding	Microsoft Corporation

16

Lab: Window Process Finder

- Use the Window process finder toolbar button to identify the owner of a window
- Lab:
  1. Open Regedit and modify HKLM\System\CurrentControlSet\Control\ ProductOptions\ProductType
  2. Move the window process finder target over the resulting popup to see what process owns the window

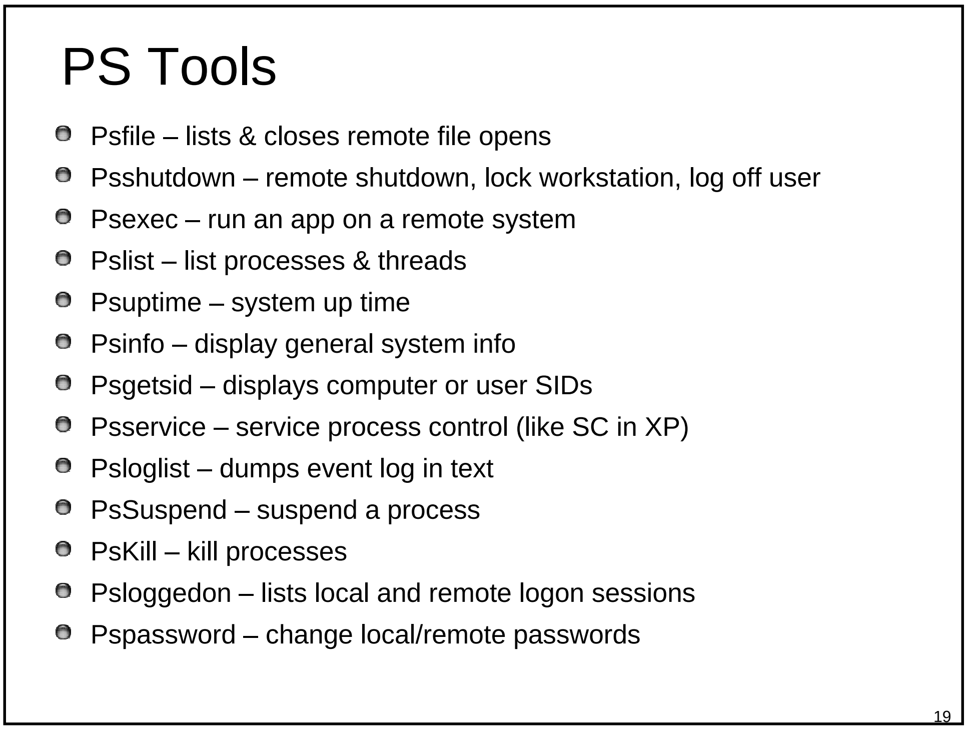
17



PS Tools

- Group of 12 process/system control tools
  - Where’d the “Ps” come from?
    - The UNIX process listing tool is named “ps”
    - The first PsTool was a UNIX “ps”-equivalent, PsList
- They all work on Windows NT4/2000/XP/2003
- They all work remotely as well as locally
  - Require admin rights to remote system
    - Can specify credentials with “-u” switch
- None require manual remote software installation

18



PS Tools

- Psfile – lists & closes remote file opens
- Psshutdown – remote shutdown, lock workstation, log off user
- Psexec – run an app on a remote system
- Pslist – list processes & threads
- Psuptime – system up time
- Psinfo – display general system info
- Psgetsid – displays computer or user SIDs
- Psservice – service process control (like SC in XP)
- Psloglist – dumps event log in text
- PsSuspend – suspend a process
- PsKill – kill processes
- Psloggedon – lists local and remote logon sessions
- Pspassword – change local/remote passwords

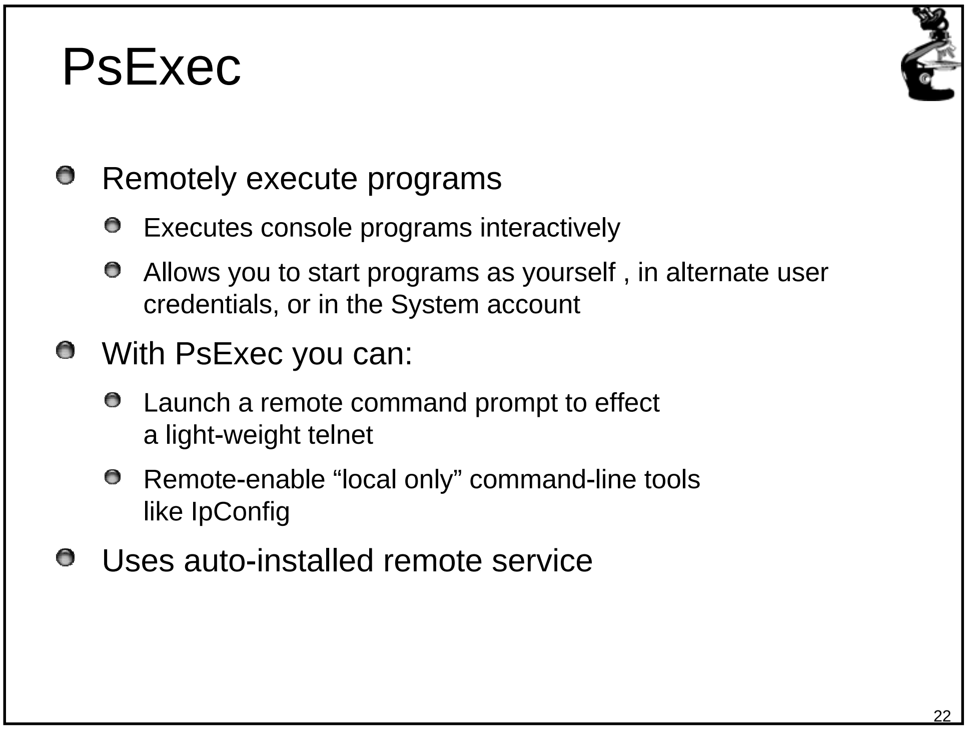
19

PsKill

- The perfect complement to PsList is PsKill
  - Similar to Resource Kit Kill and Remote Kill
  - See a process running on a remote (or local) system with PsList, kill it with PsKill
- Unlike Task Manager, PsKill lets you kill *any* process if you’re an admin
  - Uses “Debug” privilege
- Uses auto-installed remote service and TerminateProcess API

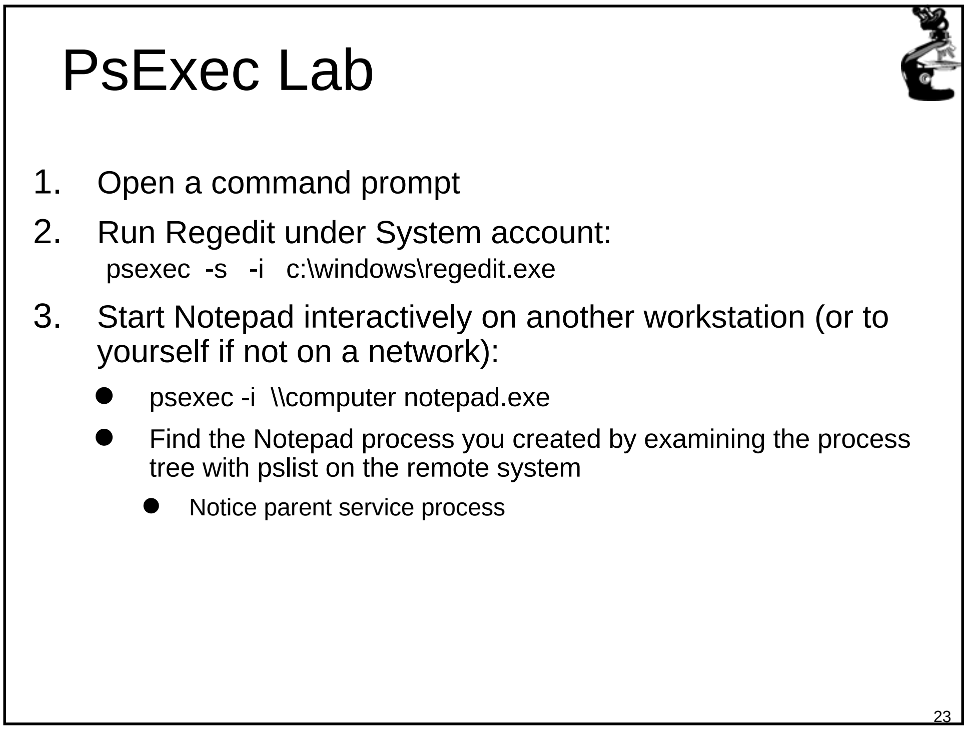
PsList/PsKill Lab

1. Open a command prompt
2. Try Pslist on your machine
   - pslist
   - pslist -t (tree view)
   - pslist -s (autorefresh)
3. Look at process list on your neighbor’s machine
   - pslist \\computername
4. Kill Explorer.exe on your neighbor’s workstation
   - pskill \\computer explorer.exe



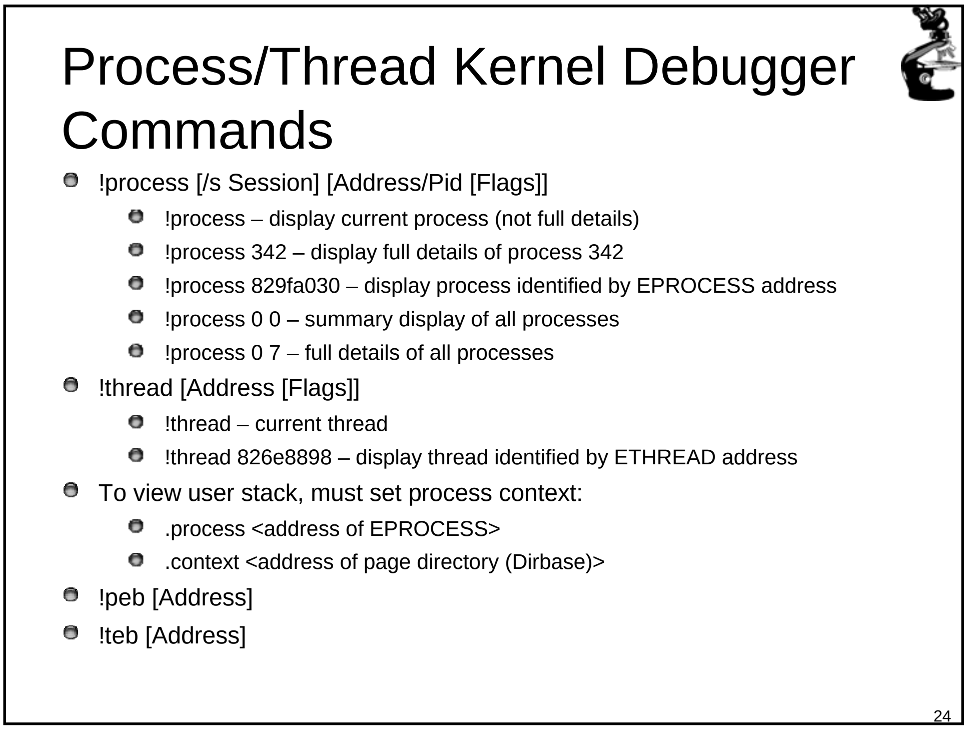
PsExec

- Remotely execute programs
  - Executes console programs interactively
  - Allows you to start programs as yourself , in alternate user credentials, or in the System account
- With PsExec you can:
  - Launch a remote command prompt to effect a light-weight telnet
  - Remote-enable “local only” command-line tools like IpConfig
- Uses auto-installed remote service



PsExec Lab

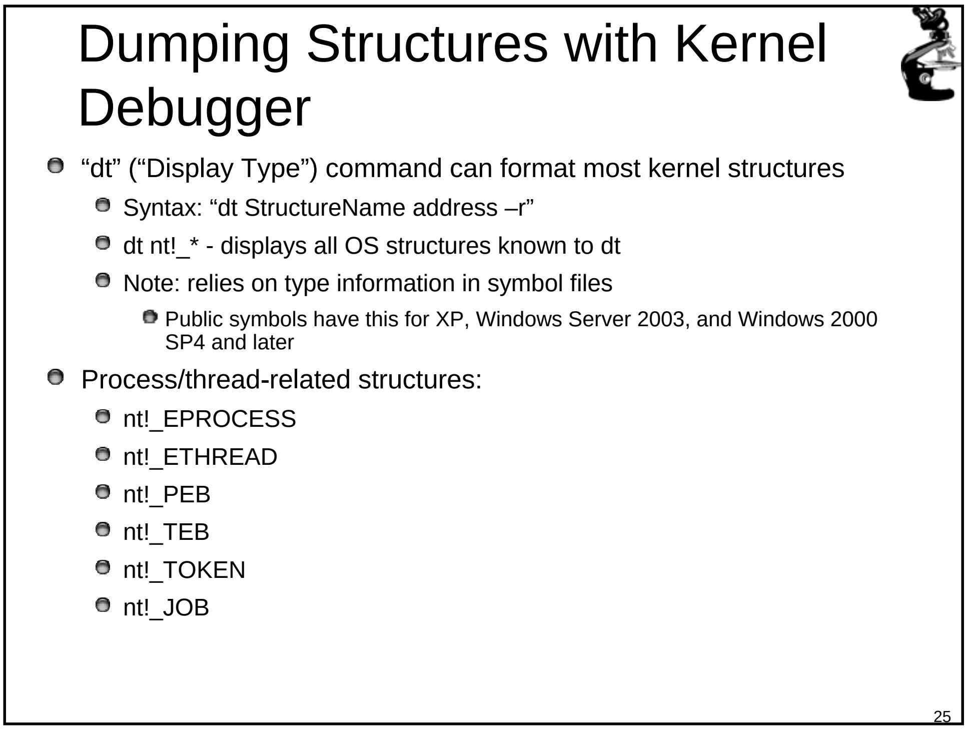
1. Open a command prompt
2. Run Regedit under System account:
   psexec -s -i c:\windows\regedit.exe
3. Start Notepad interactively on another workstation (or to yourself if not on a network):
   - psexec -i \\computer notepad.exe
   - Find the Notepad process you created by examining the process tree with pslist on the remote system
     - Notice parent service process



Process/Thread Kernel Debugger Commands

- !process [/s Session] [Address/Pid [Flags]]
  - !process – display current process (not full details)
  - !process 342 – display full details of process 342
  - !process 829fa030 – display process identified by EPROCESS address
  - !process 0 0 – summary display of all processes
  - !process 0 7 – full details of all processes
- !thread [Address [Flags]]
  - !thread – current thread
  - !thread 826e8898 – display thread identified by ETHREAD address
- To view user stack, must set process context:
  - .process <address of EPROCESS>
  - .context <address of page directory (Dirbase)>
- !peb [Address]
- !teb [Address]

24



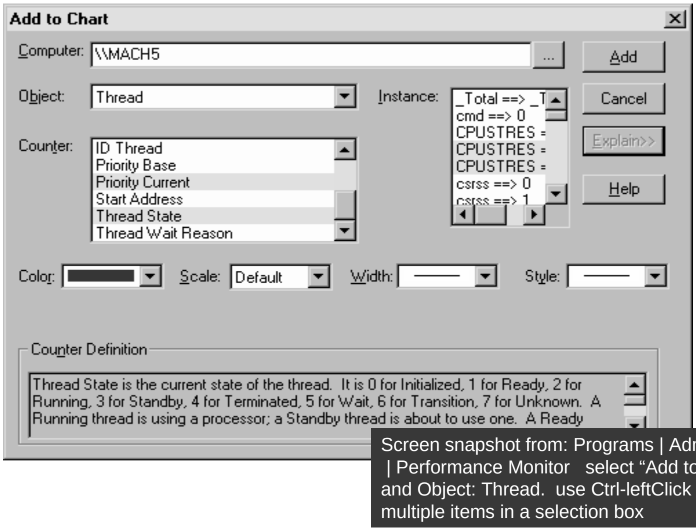
Dumping Structures with Kernel Debugger

- “dt” (“Display Type”) command can format most kernel structures
  - Syntax: “dt StructureName address –r”
  - dt nt!_* - displays all OS structures known to dt
  - Note: relies on type information in symbol files
    - Public symbols have this for XP, Windows Server 2003, and Windows 2000 SP4 and later
- Process/thread-related structures:
  - nt!_EPROCESS
  - nt!_ETHREAD
  - nt!_PEB
  - nt!_TEB
  - nt!_TOKEN
  - nt!_JOB

25

Watching the Scheduler

Performance Monitor - Threads Object



Add to Chart

Computer: \\MACH5

Object: Thread

Instance:
_Total ==> _T
cmd ==> 0
CPUSTRES =
CPUSTRES =
CPUSTRES =
csrss ==> 0
csrss ==> 1

Counter:
ID Thread
Priority Base
Priority Current
Start Address
Thread State
Thread Wait Reason

Color: ▬ Scale: Default Width: —— Style: ——

Counter Definition

Thread State is the current state of the thread. It is 0 for Initialized, 1 for Ready, 2 for Running, 3 for Standby, 4 for Terminated, 5 for Wait, 6 for Transition, 7 for Unknown. A Running thread is using a processor; a Standby thread is about to use one. A Ready

Screen snapshot from: Programs | Admin. Tools | Performance Monitor select “Add to Chart”, and Object: Thread. use Ctrl-leftClick to select multiple items in a selection box

26