4417/5417
System Administration

Lecture 3

OS Concepts

East Tennessee State University
Department of Computing
Jack Ramsey, Lecturer

Outline

32 or 64 bit?

Windows Architecture

Linux Architecture

Processes and threads

Linux / Windows

Next class

Modern OS Architecture

32 bit Vs 64 bit OS

- Processor, MB, and OS all need to support
- Number of address lines on the CPU
- 32 = 4 gigabytes
- 64 = 16 million terabytes

Move more data efficiently per clock cycle

2008 R2 / 2012 R2 only 64bit

Linux = both

Modern OS Architecture

32-bit vs. 64-bit OS and Processor

- 32-bit Windows Operating System and x86 Processor Architecture
  - Capable of addressing 4 GB of RAM
  - Each virtual machine receives 1 MB of memory and access to hardware
  - x86 uses a Complex Instruction Set Computer (CISC)
  - x86 processors use fewer registers than x64 processors
- 64-bit Windows Operating System and x64 Processor Architecture
  - Capable of addressing 128 GB of RAM
  - Enhanced performance for memory management
  - Additional security features
  - x64 architecture is backward compatible with x86
  - Process much more complex instructions at a much higher rate

ETSU

Windows

East Tennessee State University
Department of Computing
Jack Ramsey, Lecturer

Windows Architecture

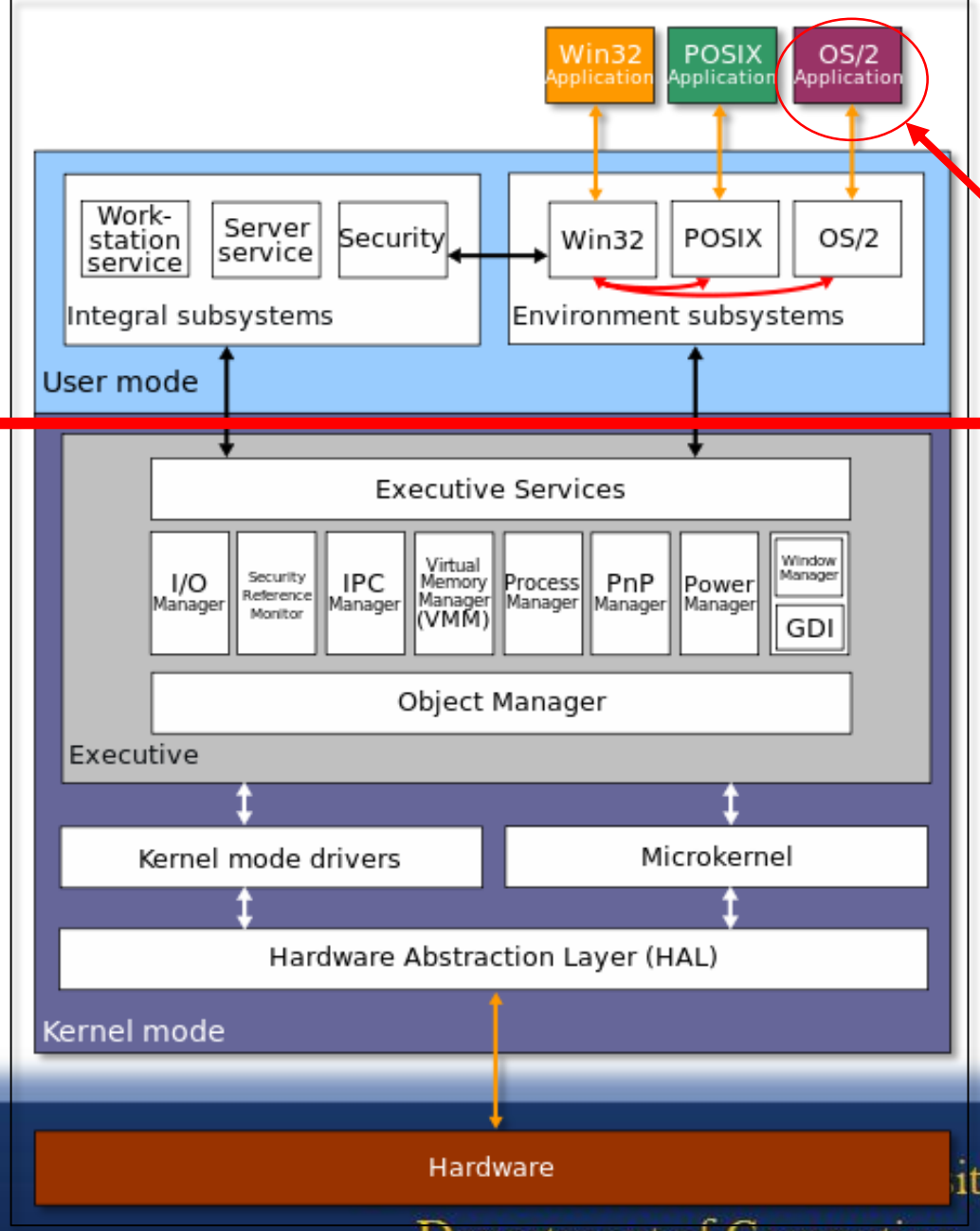
Two Major Components

- User
- Kernel

Preemptive multi-threaded architecture (eh?)

Apps -> user mode -> kernel mode -> HW

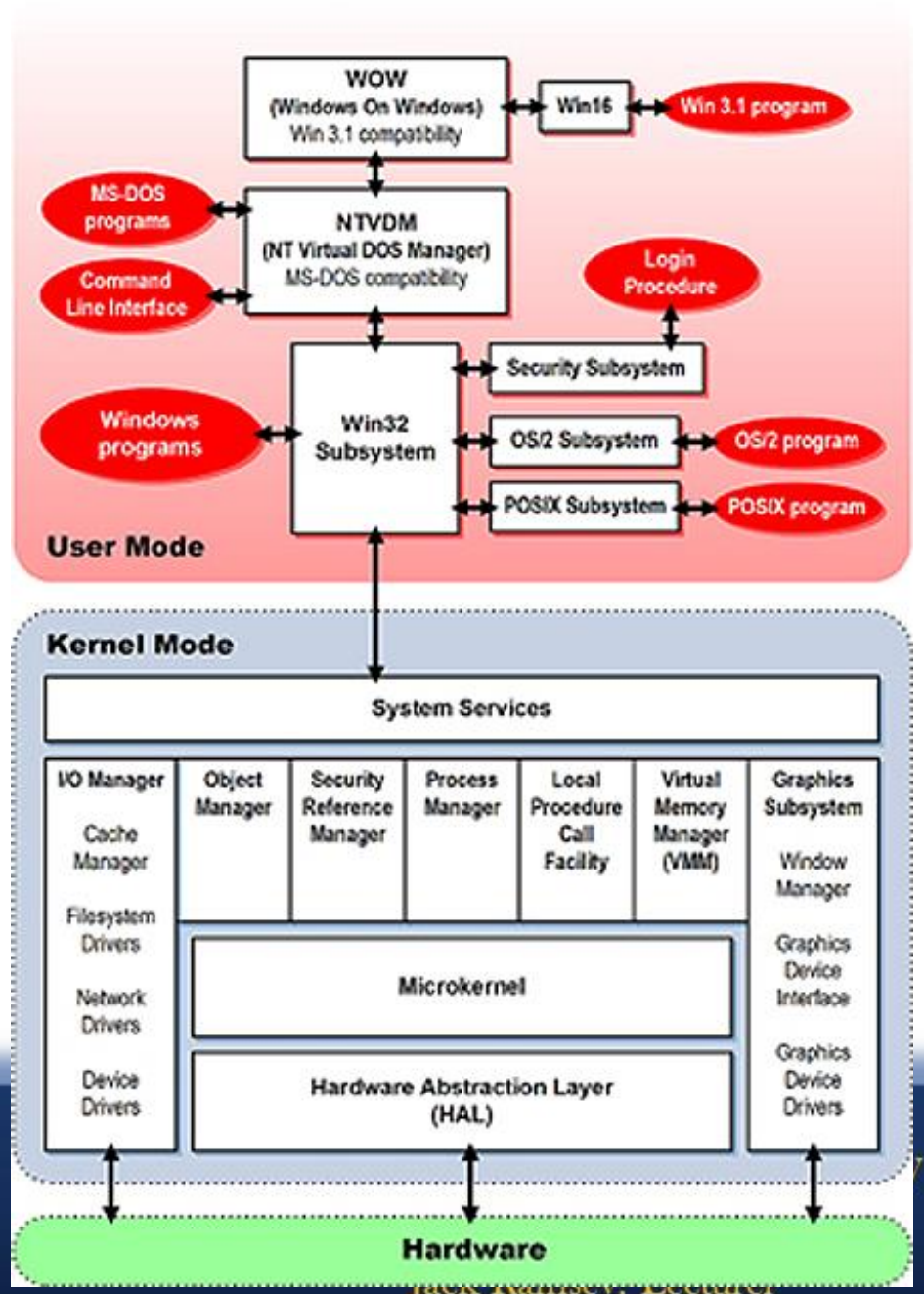
Windows Architecture



No longer present,
Starting with XP

POSIX (Portable Operating System Interface) is a set of standard operating system interfaces based on the Unix operating system.

Windows Architecture



User Mode

Applications programs run in user mode:

No direct access to hardware

Programs are limited to assigned memory address spaces

The Win32 subsystem is the primary application subsystem. All 32-bit Windows applications run in the Win32 Subsystem.

Programs use Win32's Application Program Interface (API) to request system services from a kernel mode component. This protects applications from crashing the system, and against unauthorized user access.

User Mode

DOS and Windows 16-bit applications are handled by a series of nested subsystems (culminating as always with the Win32 Subsystem)

The NT Virtual DOS Machine (NTVDM) provides a DOS-compatible environment for DOS programs

16-bit Windows program communicate first with a subsystem designed to handle such applications

16-bit system calls; these calls are converted to the 32-bit calls used by Windows NT in a subsystem called Windows on Win32 (WOW).

These applications also require a NTVDM environment because they also depend on DOS services

User Mode

In 64-bit versions of Windows the main subsystem is Win64

A Windows-on-Win64 (WoW64) subsystem allows 32-bit programs to interact with the 64-bit Windows executive. Thus programs designed for Win32 can run on Win64

The OS/2 and Posix subsystems allow Windows to run programs built for OS/2 or Posix operating systems, where supported

Security Subsystem supports the logon process. The Security Subsystem also communicates with the Win32 Subsystem

Kernel Mode

All code that runs in kernel mode can:

Access the hardware directly

Access all memory.

The entire set of services that comprise kernel mode is called Executive Services (or sometimes the Windows NT Executive).

The I/O Manager controls most input and output on the system.

The Object Manager creates, modifies and deletes system objects. These objects represent a specific instance of a resource (for example, a file, a process, or a port).

Kernel Mode

The Security Reference Manager (SRM) is responsible for enforcing system security settings by granting or denying access to objects and system resources upon request from the Object Manager. This process relies on data structures known as security access tokens (SATs)

The Process Manager creates and manages system processes. However, process scheduling is handled by the microkernel

The Local Procedure Call Facility is responsible for communication between processes

Kernel Mode

The Virtual Memory Manager handles the allocation and use of the system's memory. Virtual memory is the physical space on a hard disk that NT treats as though it were RAM. Virtual memory can also be thought of as an extension of RAM, or "fake" RAM. Memory is divided into 'pages' and is stored in a pagefile on disk

Window Manager is responsible for providing all of the GUI. It communicates directly with the Graphics Device Drivers, which in turn communicate directly with hardware

The five other kernel mode subsystems communicate directly with the microkernel, the very heart of the NT operating system. It handles interrupts, schedules threads, and synchronizes processing activity. The microkernel, in turn, communicates with the hardware abstraction layer (HAL)

The Kernel

Handles basic I/O

Interfaces between Executive Services and HW

ntoskrnl.exe

64 bit Windows

64 bit OS, 64 bit CPU, 64 bit App = native

64 bit OS, 64 bit CPU, 32 bit App = VM

WOW64 (Windows On Windows)

Creates mini VMs for each 32 bit app on a 64 bit OS

Wow64.dll is loaded to be the middle man between the 32 bit app and the 64 bit OS

Linux

East Tennessee State University
Department of Computing
Jack Ramsey, Lecturer

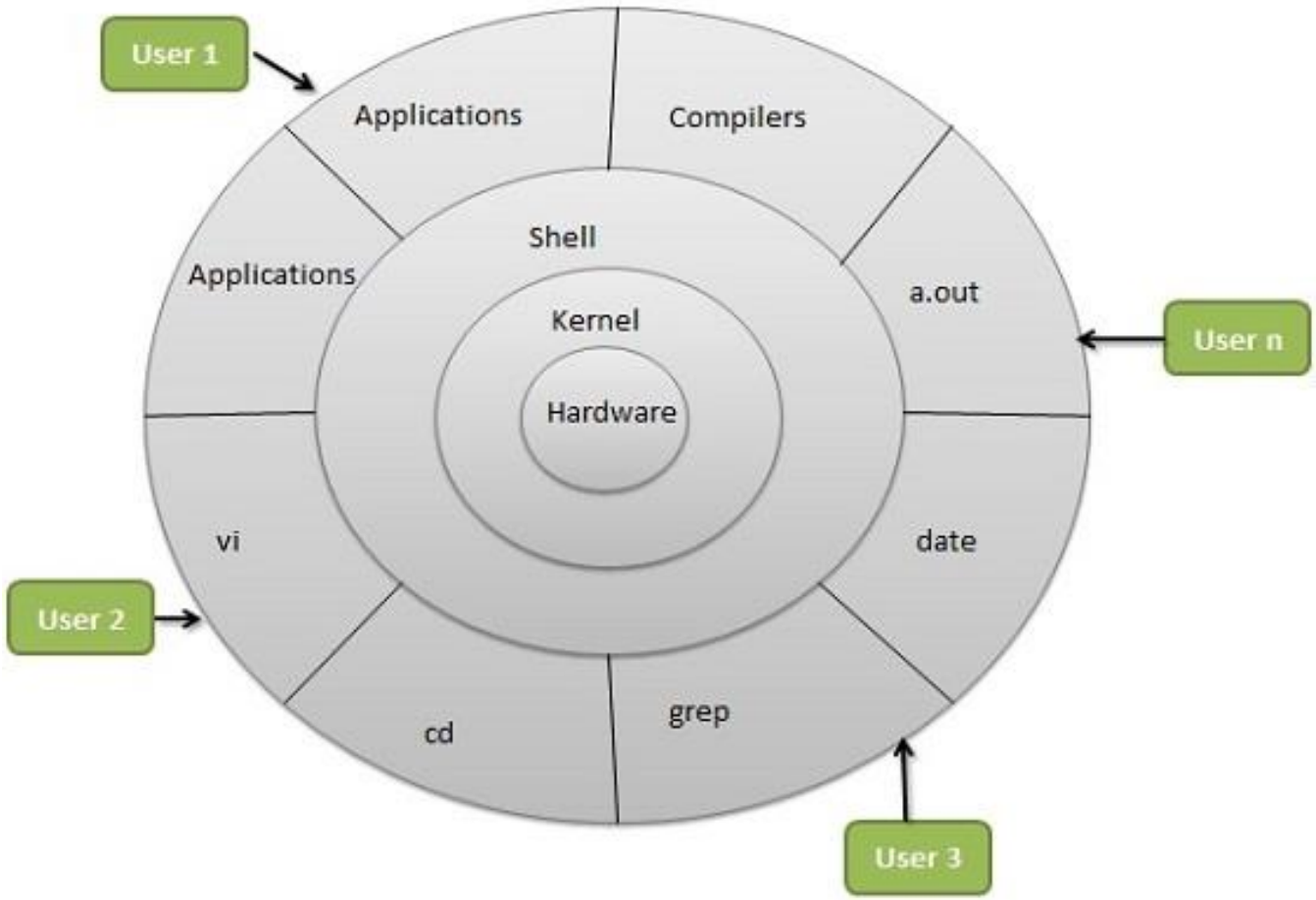
A View from Linux

Hardware layer - Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).

Kernel - Core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.

Shell - An interface to kernel, hiding complexity of kernel's functions from users. Takes commands from user and executes kernel's functions.

Utilities - Utility programs giving user most of the functionalities of an operating system.



* http://www.tutorialspoint.com/operating_system/os_linux.htm

Linux Features

Portable - software works on different types of hardware in same

Open Source - Linux source code is freely available and it is community based development project.

Multi-User - multiple users can access system resources like memory/ ram/ application programs at same time.

Multiprogramming - multiple applications can run at same time.

* http://www.tutorialspoint.com/operating_system/os_linux.htm

Linux Features

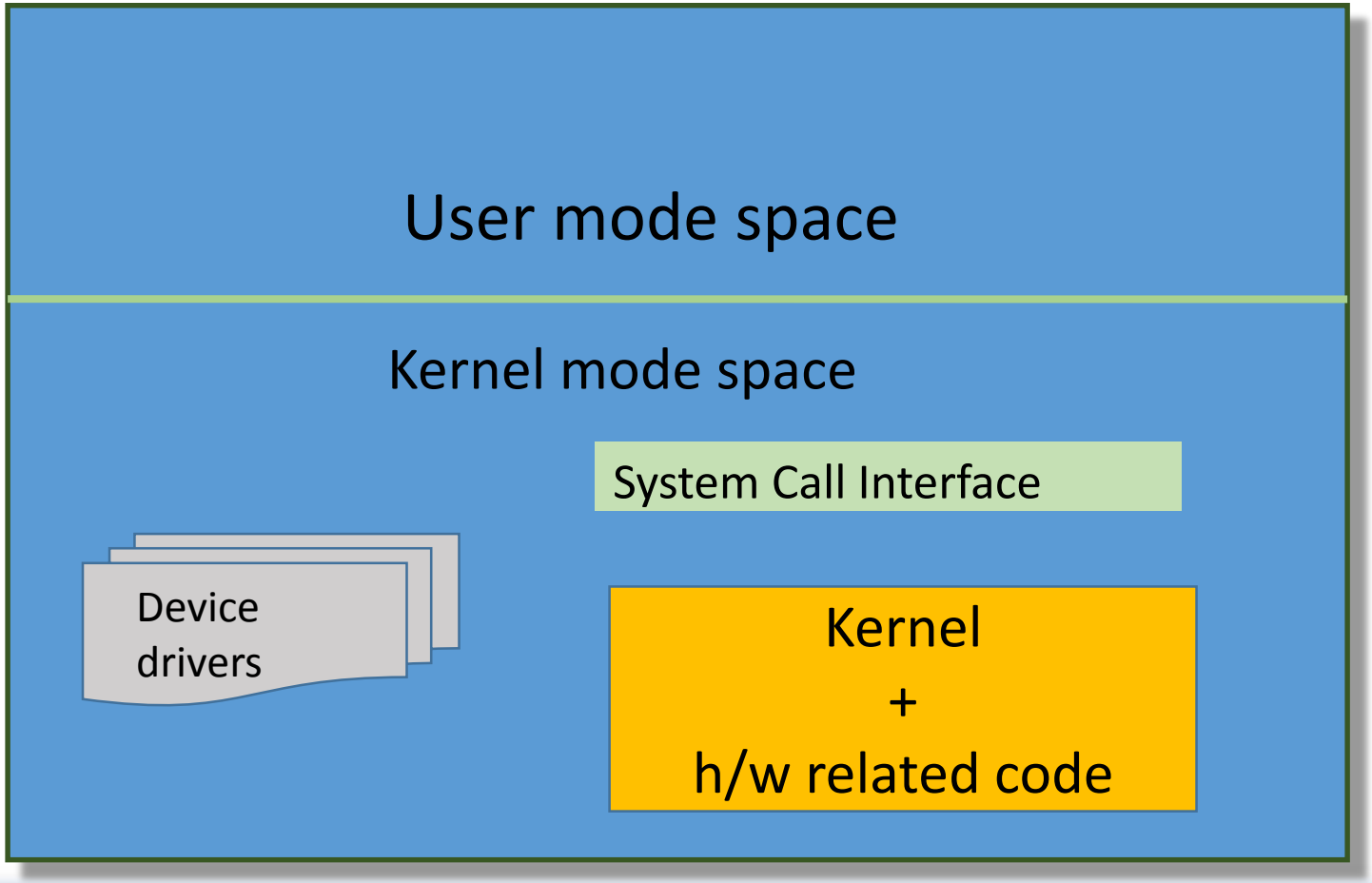
Hierarchical File System - standard file structure in which system files/ user files are arranged.

Shell - Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs etc.

Security - Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

* http://www.tutorialspoint.com/operating_system/os_linux.htm

System Structure (Simplified)



Processes

East Tennessee State University
Department of Computing
Jack Ramsey, Lecturer

Processes

Program vs. Process

- A program is a static sequence of instructions
- A process is a container for a set of resources used to execute a program*

*Russinovich, M. & Margosis, A. (2011). *Windows Sysinternals Administrator's Reference*. Redmond, WA: Microsoft Press.

Processes

Process is comprised of address space and data (instructions)

Address spaces allocated in pages

Pages may reside in virtual memory (nothing to do with VMware Player) and physical memory

Virtual memory referred to as swap space or pagefile

Processes

Virtual memory is a feature of an operating system (OS) that allows a computer to compensate for shortages of physical **memory** by temporarily transferring pages of data from random access **memory** (RAM) to disk storage.*

This is referred to as the swap space (vs. Windows' pagefile)

Separate partition on disk

*http://searchstorage.techtarget.com/definition/virtual-memory

Processes

Process MAY = application

More accurately the part of an application that is running

Processes

The kernel keeps track of the following:

- Processes address space
- Current state of process (sleeping, running, etc)
- Priority
- Signal mask
- Files and ports in use
- Resource list
- ID
- ***owner***

Threads

Part that runs on the processor

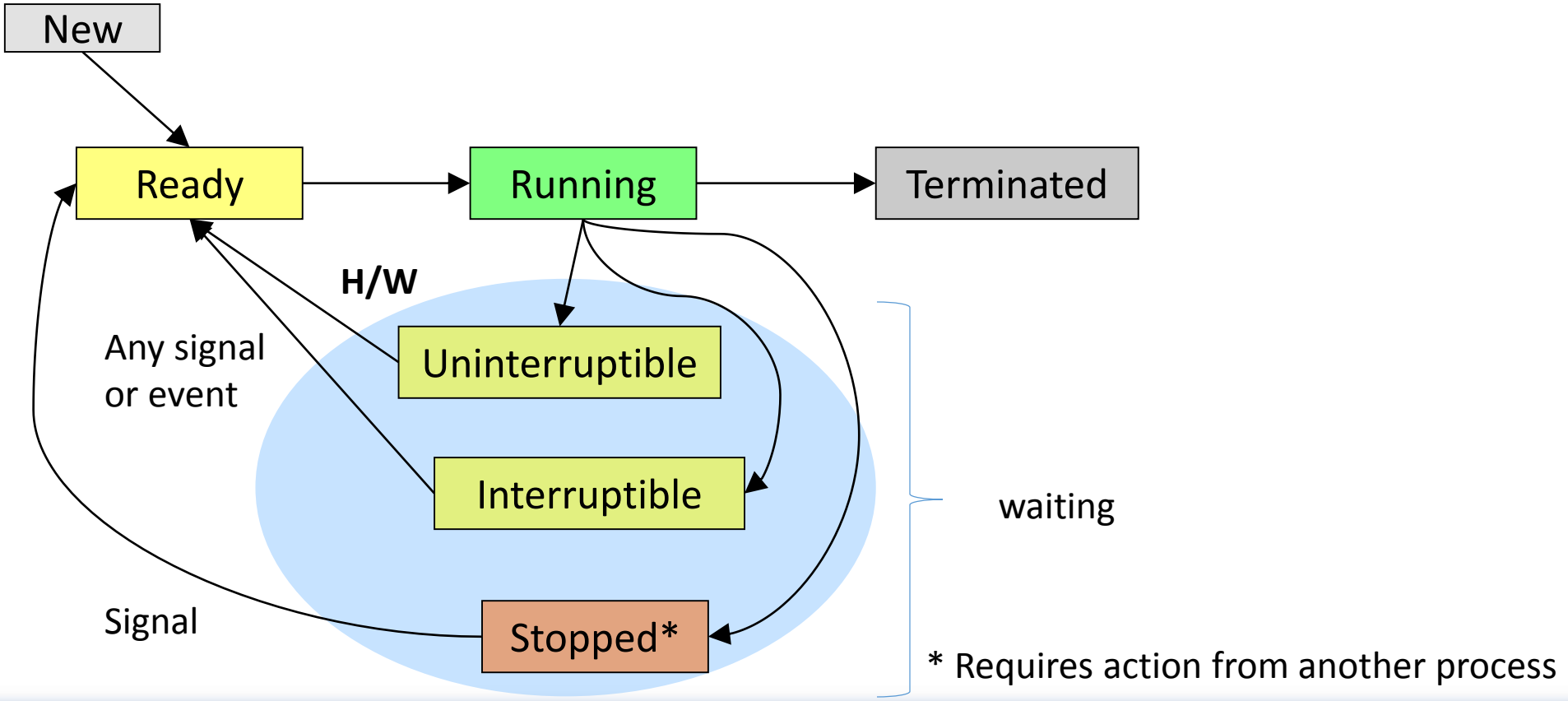
Processes fork threads to accomplish specific tasks

Inherits many attributes of the parent process

Multiple threads can run in parallel, hence

“Multithreaded OS”

Linux Process/Thread States



`fork`

1 process can spawn another process by issuing the `fork` command

The resulting child will the issue an `exec` command to start execution

`init` (SysV)

At system boot, in Linux, the kernel creates and installs the init process

PID = 1

Execute startup scripts

Similar to startup folder or other initialization registry entries in Windows

`init` (SysV)

Continues to run after startup

Handles calls to shut down or reboot

“As time went on, it became clear that init was getting too slow and inflexible for today’s computers.” *

*http://www.zdnet.com/article/after-linux-civil-war-ubuntu-to-adopt-systemd/

Upstart - 2006

Canonical (the company that manages Ubuntu), created Upstart

Event-based replacement

Handles starting of tasks & services during boot, stopping them during shutdown & supervising them while running*

Last version release: 4 September 2014

*http://upstart.ubuntu.com/

systemd - 2010

freedesktop.org*

Starts system resources in parallel, instead of running a series of scripts

Supports System-V startup scripts

Same services as System-V, but faster

*http://www.freedesktop.org/wiki/Software/systemd/

systemd - 2010

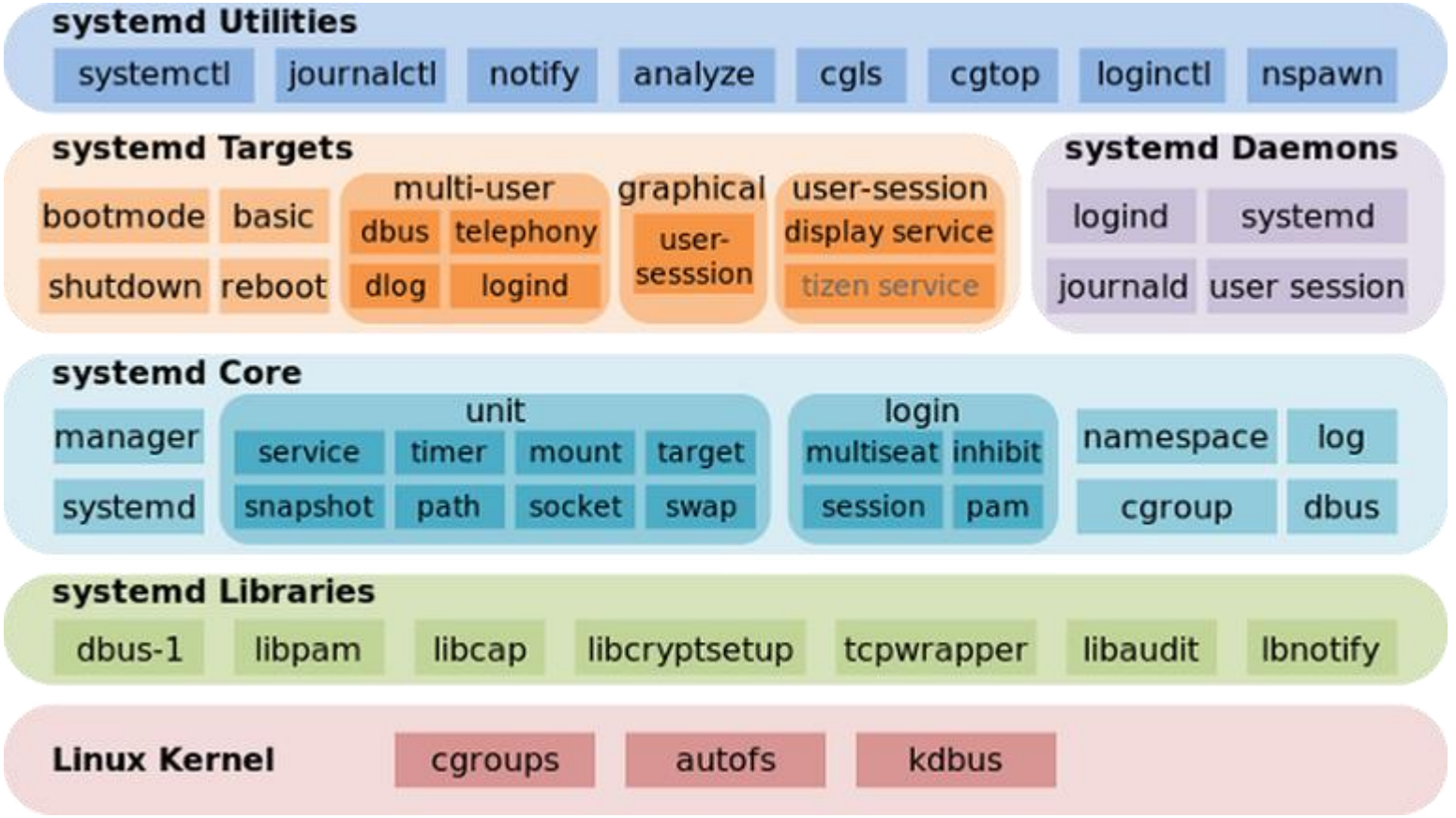
Debate between Upstart supporters (Canonical wanted to make it the Linux standard) became what ZDNet describes as “heated” *

Eventually Canonical threw in the towel in favor of systemd

systemd is currently available as an option on Ubuntu (since 15.04)

*http://www.zdnet.com/article/after-linux-civil-war-ubuntu-to-adopt-systemd/

systemd - 2010



*

*http://www.zdnet.com/article/after-linux-civil-war-ubuntu-to-adopt-systemd/

Signals

Used for

- Process to process communication
- Kill, interrupt, suspend processes
- Admin kill
- Programming errors
- Notification of interesting occurrences within the OS
  - Death of a child
  - I/O channel is available

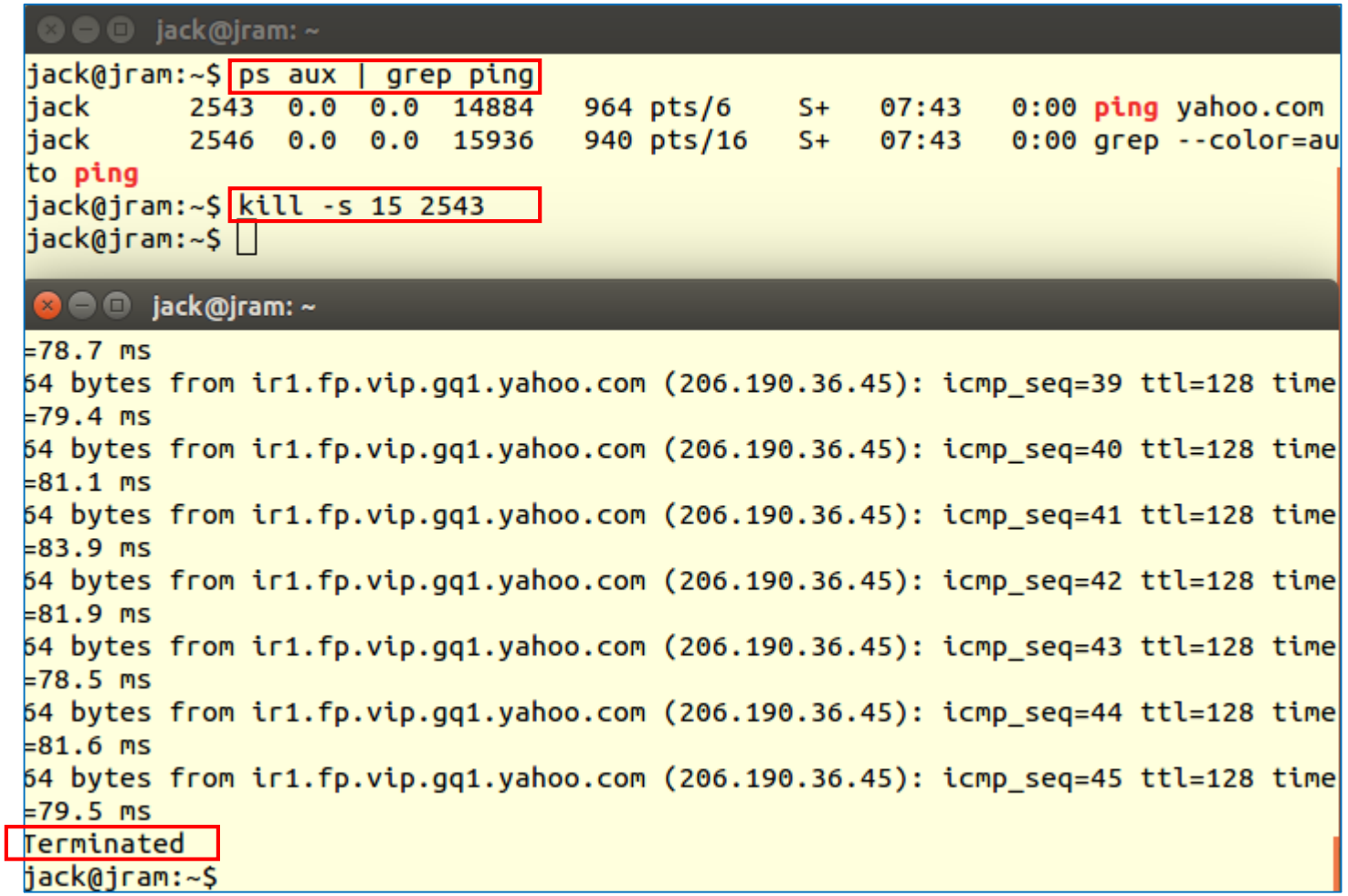
Ending a Process

Linux

`kill -9` *`pid`*

Not graceful (SIGTERM much more graceful), i.e.,

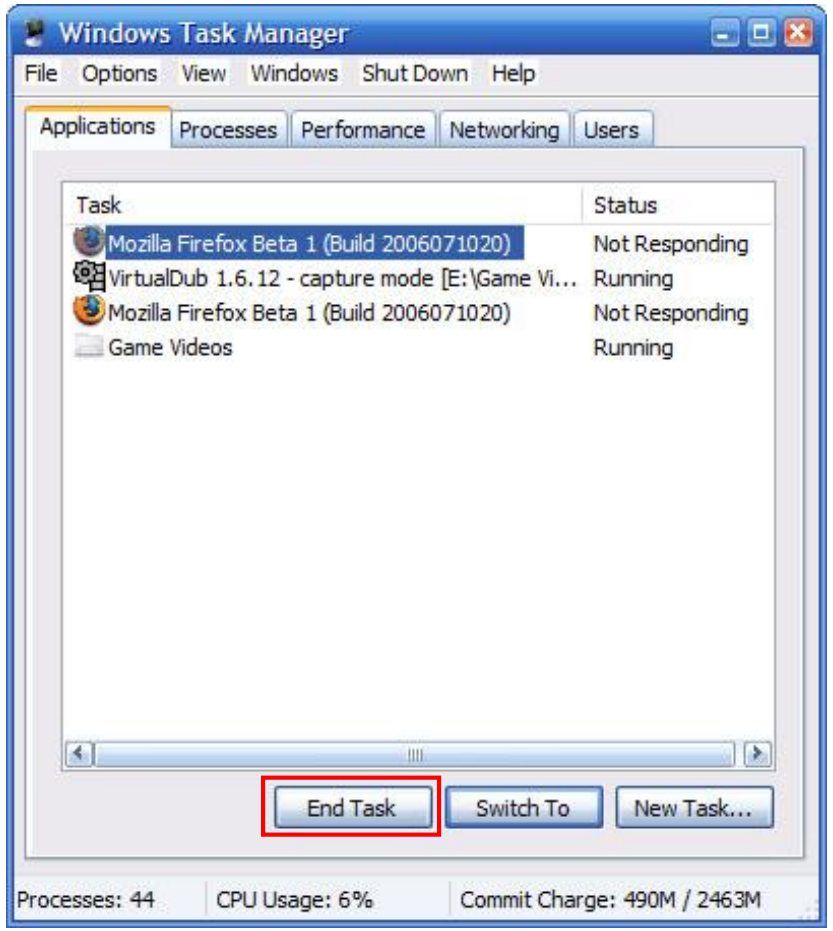
kill -s 15 *`pid`*



Ending a Process

Windows Task Manager

Process Explorer provides more fine-grained control than TM



Listing Processes

Linux

`ps aux`

- `a` = show all processes
- `u` = user orientated format
- `x` = no control terminal

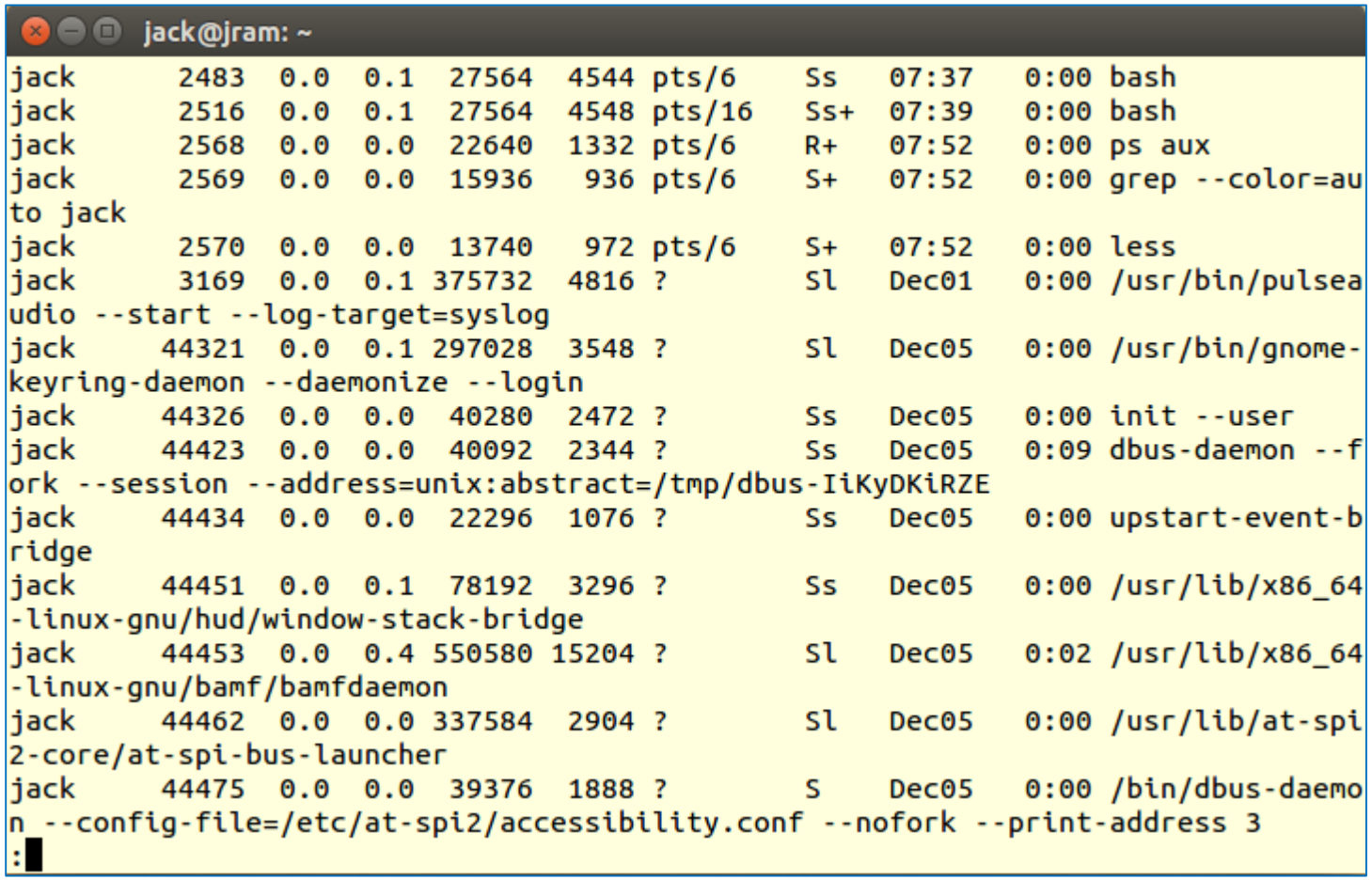
jack@jram: ~

```
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.0  33912  2896 ?        Ss   Dec01   0:03 /sbin/init
root          2  0.0  0.0      0     0 ?        S    Dec01   0:00 [kthreadd]
root          3  0.0  0.0      0     0 ?        S    Dec01   0:08 [ksoftirqd/0]
root          4  0.0  0.0      0     0 ?        S    Dec01   0:00 [kworker/0:0]
root          5  0.0  0.0      0     0 ?        S<   Dec01   0:00 [kworker/0:0H]
root          7  0.0  0.0      0     0 ?        S    Dec01   1:52 [rcu_sched]
root          8  0.0  0.0      0     0 ?        R    Dec01   1:50 [rcuos/0]
root          9  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/1]
root         10  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/2]
root         11  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/3]
root         12  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/4]
root         13  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/5]
root         14  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/6]
root         15  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/7]
root         16  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/8]
root         17  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/9]
root         18  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/10]
root         19  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/11]
root         20  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/12]
root         21  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/13]
root         22  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/14]
root         23  0.0  0.0      0     0 ?        S    Dec01   0:00 [rcuos/15]
:
```

Listing Processes

Linux

Processes owned by a given user



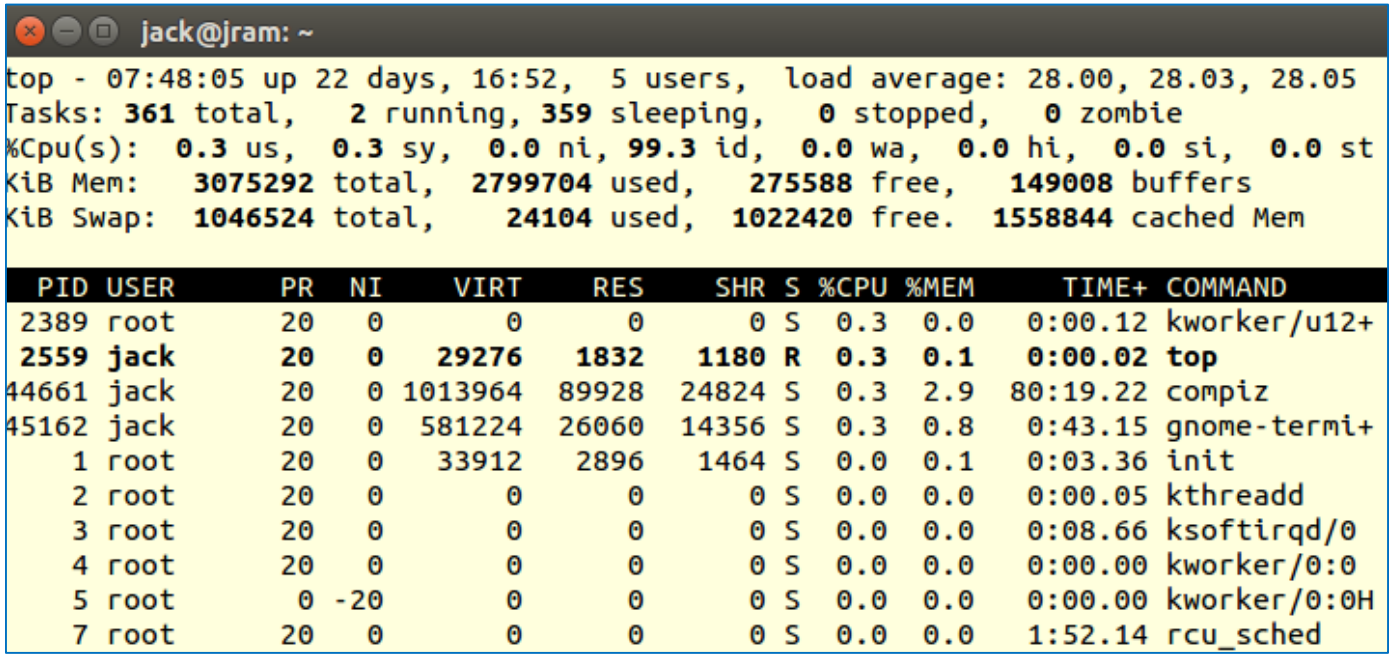
`ps aux | grep jack | less`

Process Monitoring

`top`

Unlike `ps`, `top` auto-updates to provide real-time monitoring information

Several options available while running (press ‘h’ for a list).



/proc

Logical directory maintained by the system

Apps can query it to get system state information

File usually have zero size because it is built on the fly

jack@jram: ~

```
dr-xr-xr-x   9 root       root                0 Dec 24 07:37 2389/
dr-xr-xr-x   9 rtkit      rtkit               0 Sep 10 11:11 2391/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 24/
dr-xr-xr-x   9 root       root                0 Dec 24 07:37 2407/
dr-xr-xr-x   9 root       root                0 Dec 24 07:37 2408/
dr-xr-xr-x   9 root       root                0 Dec 24 07:37 2416/
dr-xr-xr-x   9 root       root                0 Dec 24 07:37 2468/
dr-xr-xr-x   9 root       root                0 Dec 24 07:37 2477/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 248/
dr-xr-xr-x   9 jack       jack                0 Dec 24 07:37 2483/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 249/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 25/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 250/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 251/
dr-xr-xr-x   9 jack       jack                0 Dec 24 07:39 2516/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 252/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 253/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 254/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 255/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 256/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 257/
dr-xr-xr-x   9 root       root                0 Sep 10 11:10 258/
dr-xr-xr-x   9 jack       jack                0 Dec 24 07:56 2588/
:
```

Output from `ll /proc | less`

Other Process Killers

`pkill` - kill process by its name, user name, group name, terminal, UID, and GID

e.g., **`pkill -s 15 -u jack firefox`**

`killall` - sends the signal to all processes.

e.g., **`sudo killall -s 15 httpd`**

Process Explorer

The following material is from Chapter 3, Russinovich, M. & Margosis, A. (2011). *Windows Sysinternals Administrator's Reference*. Redmond, WA: Microsoft Press.

East Tennessee State University
Department of Computing
Jack Ramsey, Lecturer

Handles

Windows kernel-mode consists of various subsystems (e.g., Process Manager)

Each defines one or more objects to represent resources they expose to applications (e.g., Pocess objects)

*Russinovich, M. & Margosis, A. (2011). *Windows Sysinternals Administrator's Reference*. Redmond, WA: Microsoft Press.

Handles

When an application wants to use a resource, it must call the appropriate API to create or open the resource (e.g., CreateFile function opens or creates a file)

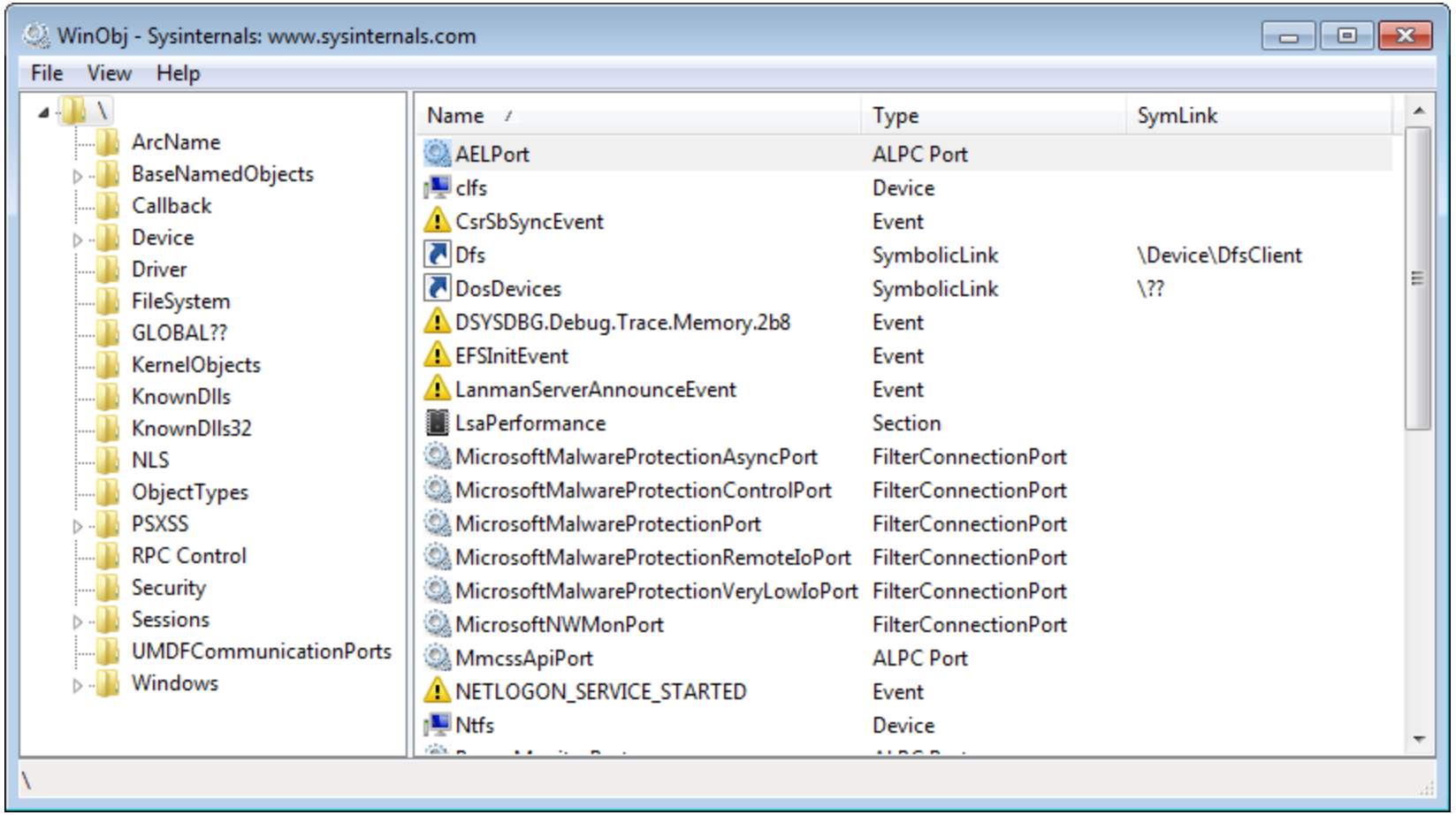
If successful, Windows allocates a reference to the object

*Russinovich, M. & Margosis, A. (2011). *Windows Sysinternals Administrator's Reference*. Redmond, WA: Microsoft Press.

Handles

WinObj

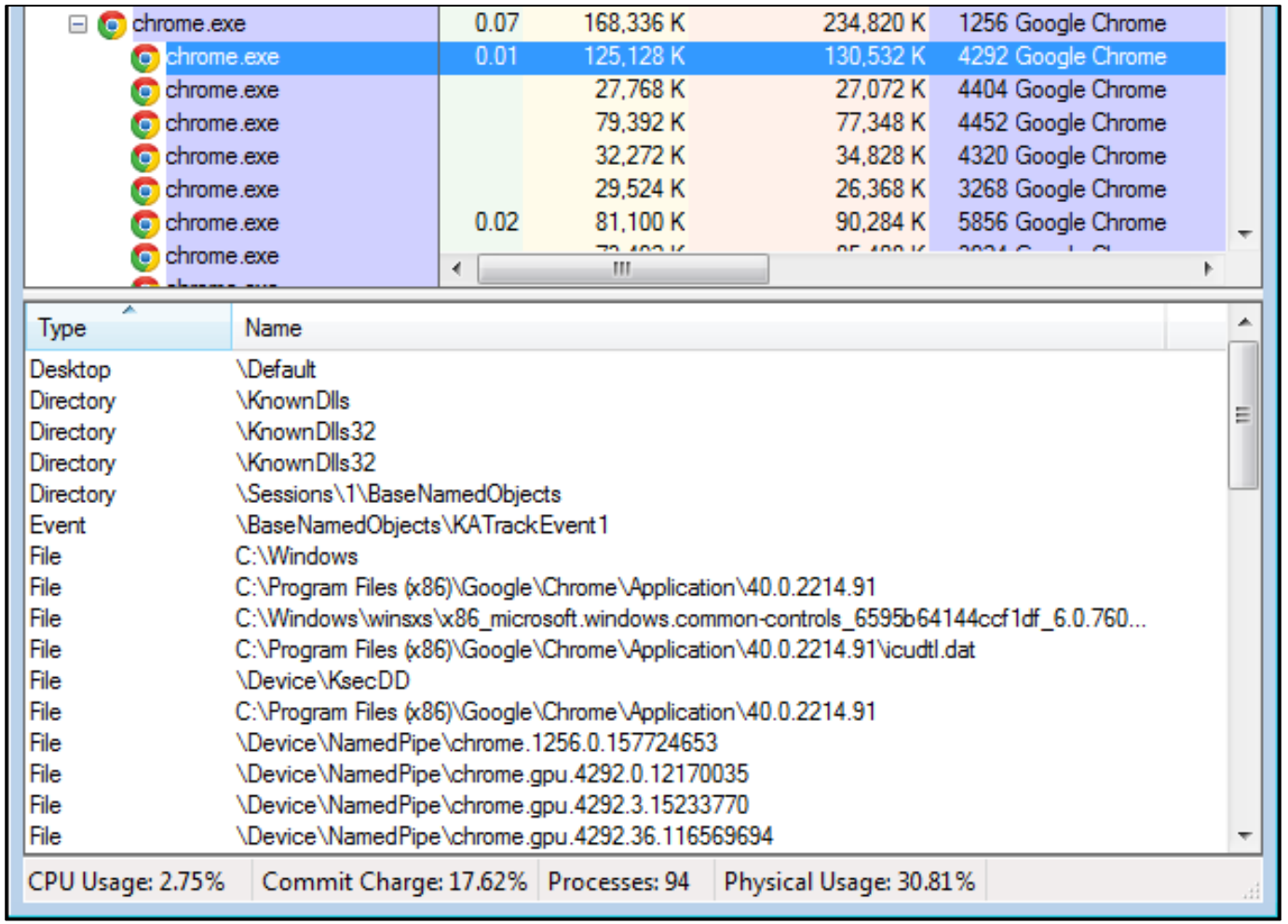
Displays Object types that Windows defines



*Russinovich, M. & Margosis, A. (2011). *Windows Sysinternals Administrator's Reference*. Redmond, WA: Microsoft Press.

Handles

Process Explorer can display the handles associated with a given process



*Russinovich, M. & Margosis, A. (2011). *Windows Sysinternals Administrator's Reference*. Redmond, WA: Microsoft Press.

Jobs

Windows provides an extension to the process model called a job

Allows groups of processes to be managed and manipulated as a unit

Example:

> A job can be used to terminate a group of processes all at once, rather than one at a time & without the calling process having to know which processes are in the group

*Russinovich, M. & Margosis, A. (2011). *Windows Sysinternals Administrator's Reference*. Redmond, WA: Microsoft Press.

Jobs

Allows control of certain attributes

Provides limits for the process or processes associated with the job.

*Russinovich, M. & Margosis, A. (2011). *Windows Sysinternals Administrator's Reference*. Redmond, WA: Microsoft Press.

Process Explorer

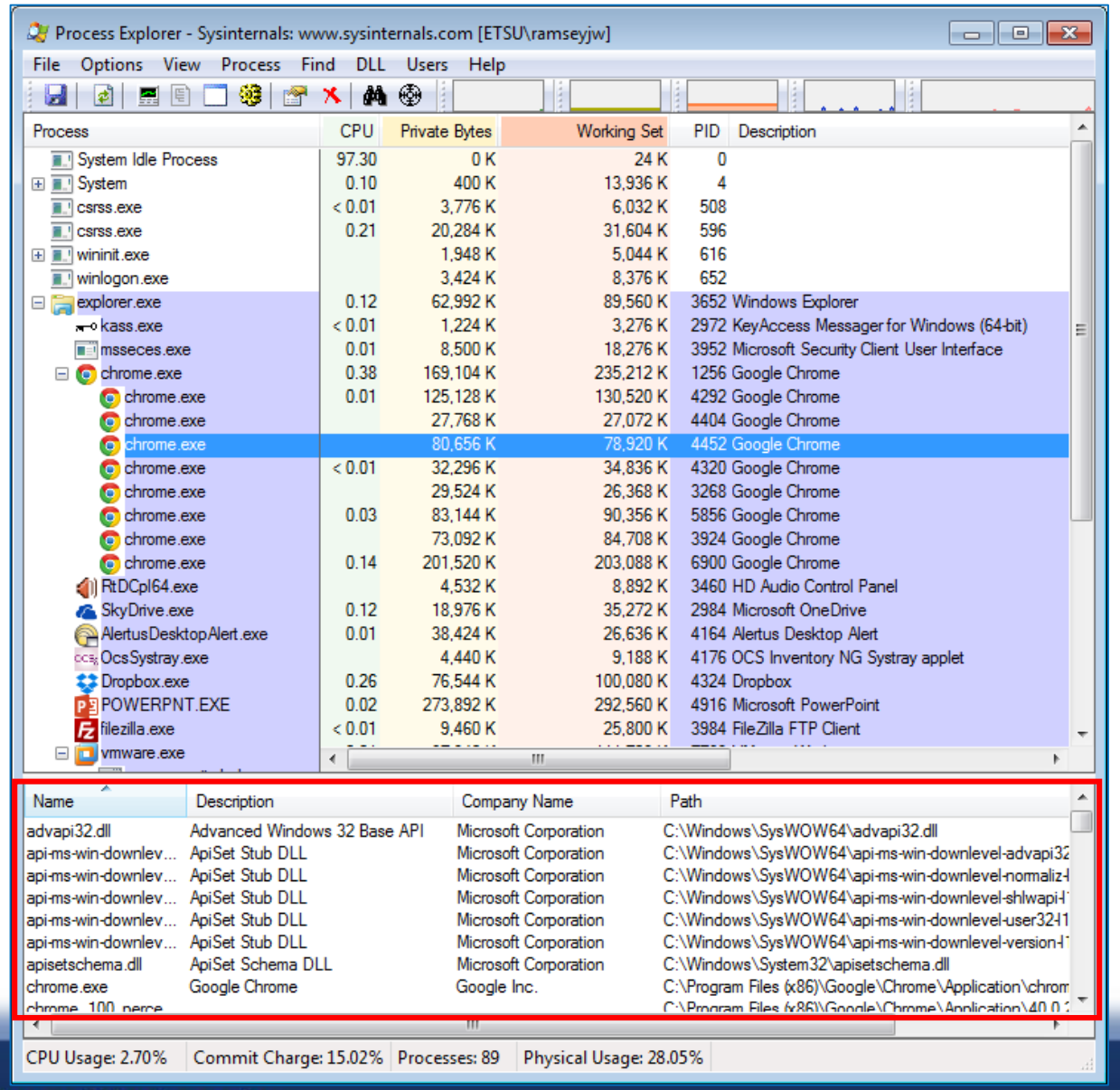
If possible, Run as Administrator

View -> Show lower pane

Ctrl+D: Lists DLLs

Ctrl+H: Lists Handles

Ctrl+L: Toggles lower pane open or closed

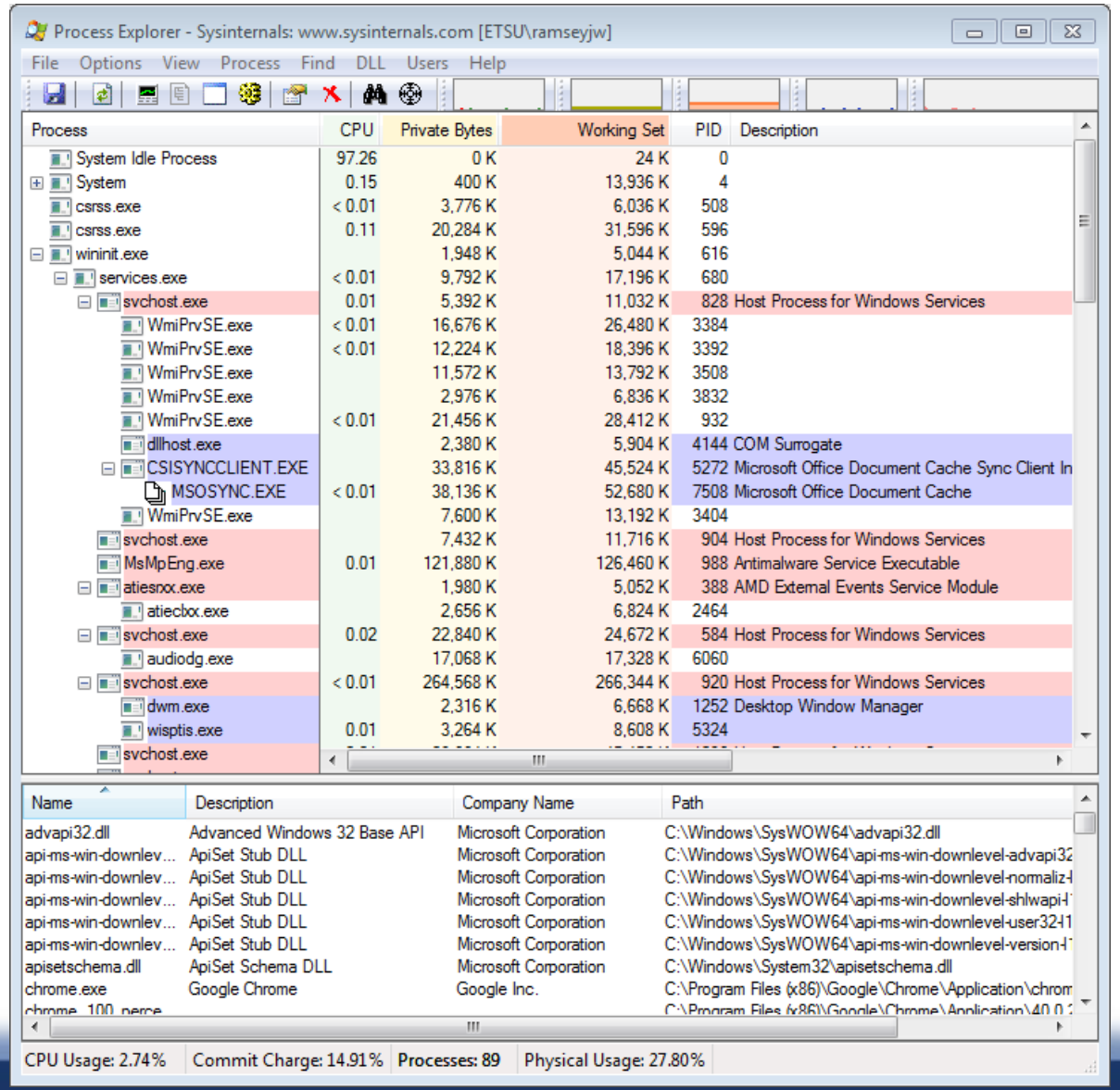


Process Explorer

Top Pane

Processes arranged in parent/child format, displaying which processes ‘own’ others

CPU time more accurate than Task Manager



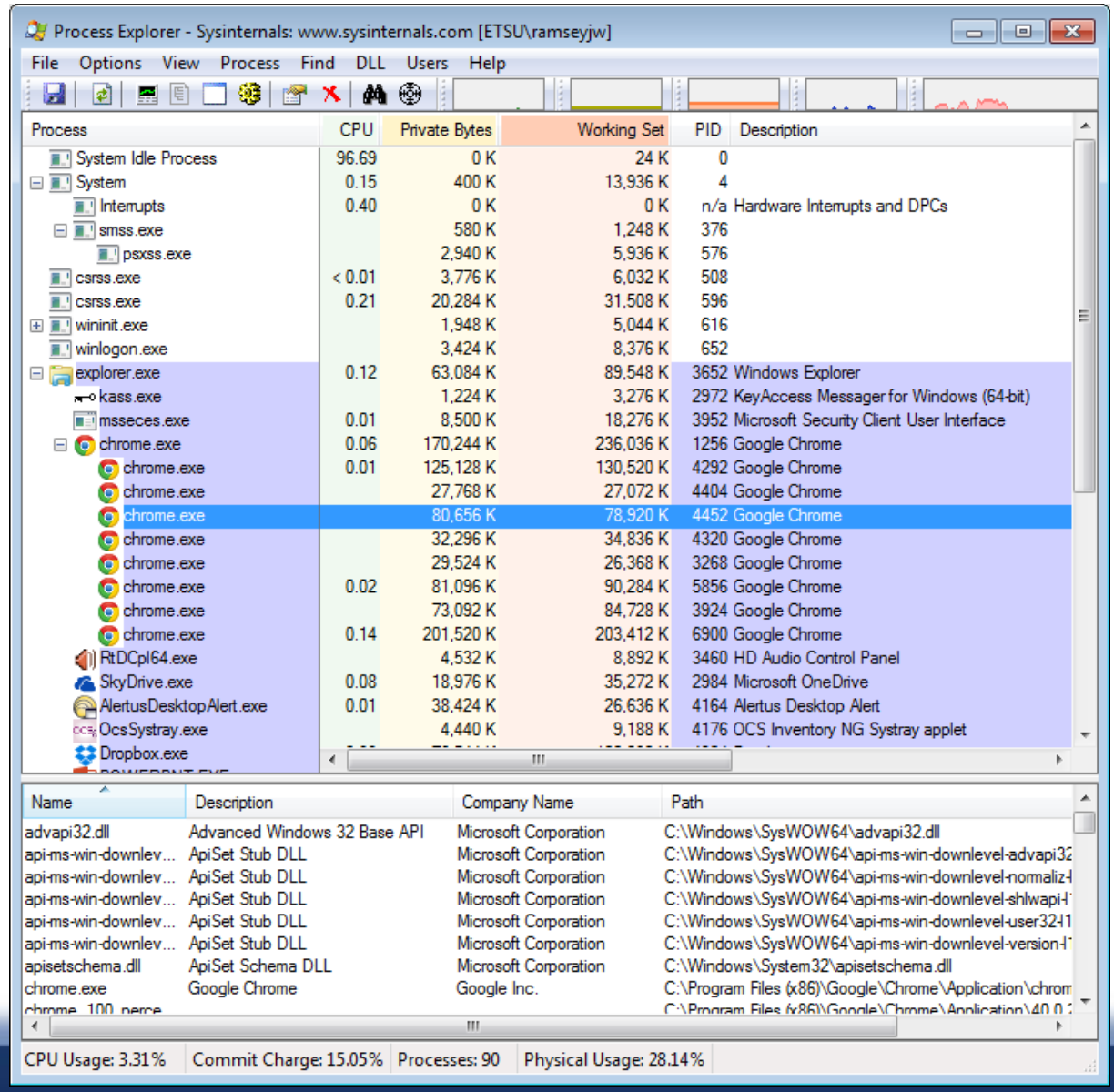
Process Explorer

Top half (collapsed on pic) represents system-level processes

1st three aren't actual processes

System Idle used to account for unused CPU time

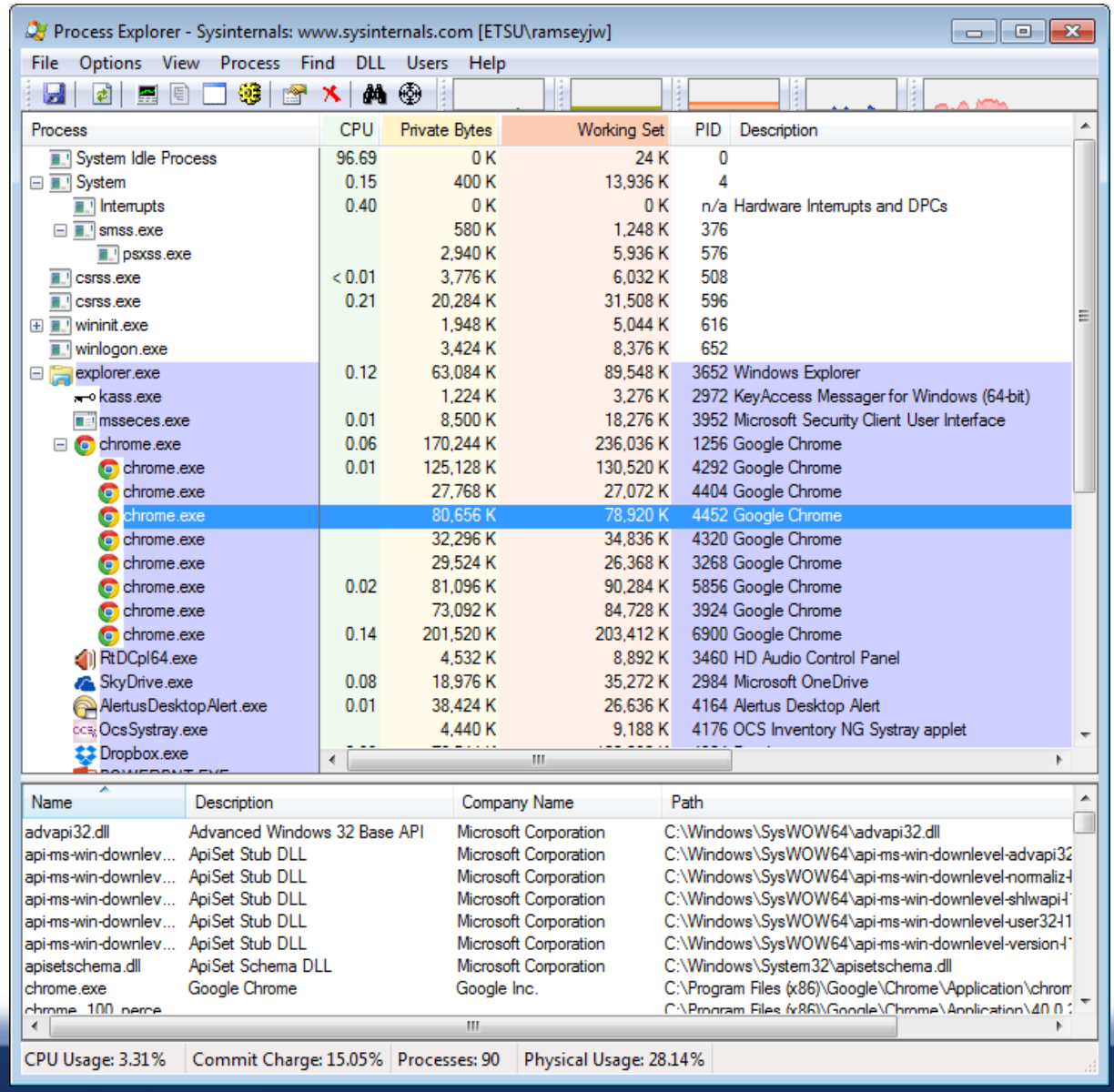
System hosts only kernel-mode threads



Process Explorer

User shell (explorer.exe)

Hosts user applications

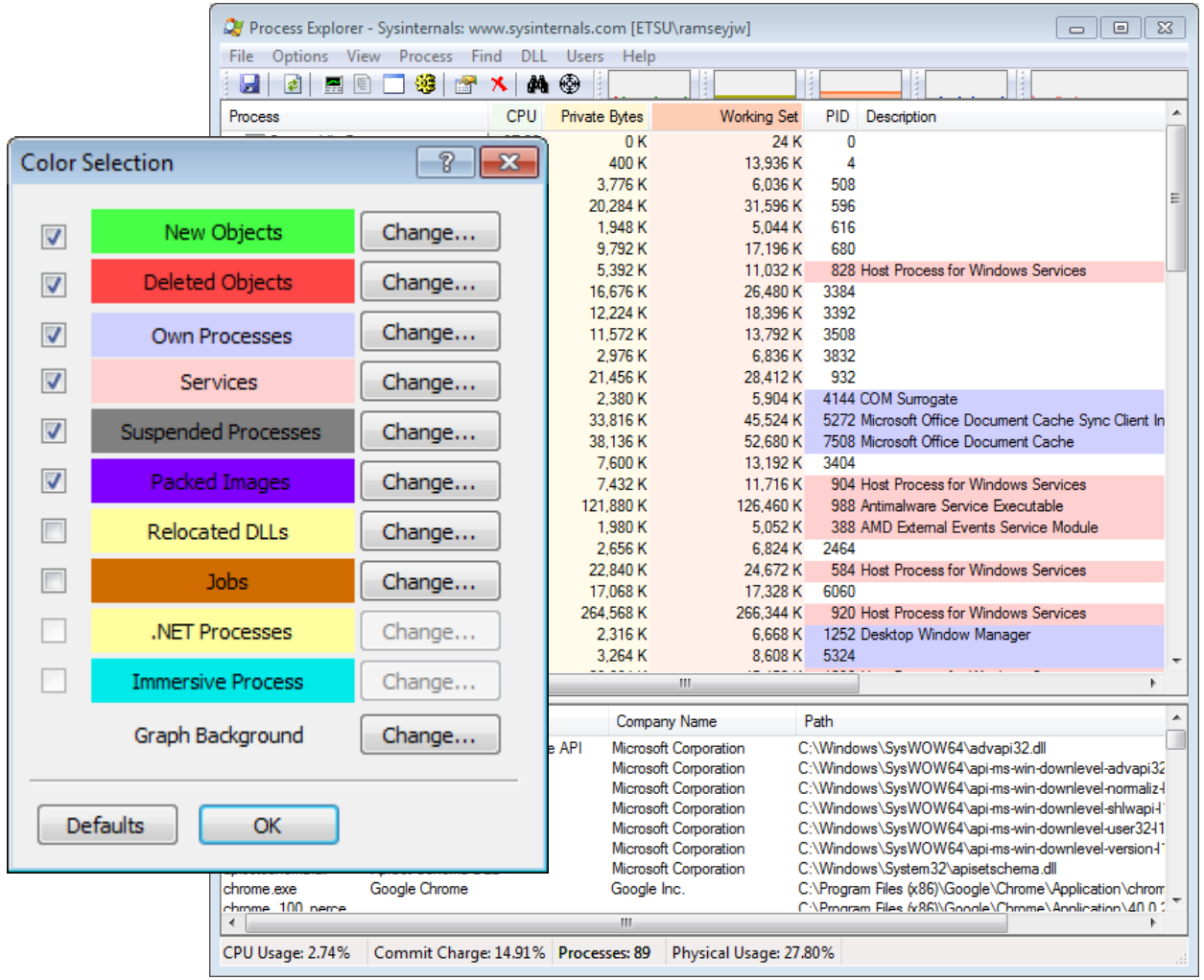


Process Explorer

Color Coding

- Distinguishes different types of processes
- Configurable (Default shown)
- Green = new process
- Red = process terminated

Access: Options -> Configure Colors…

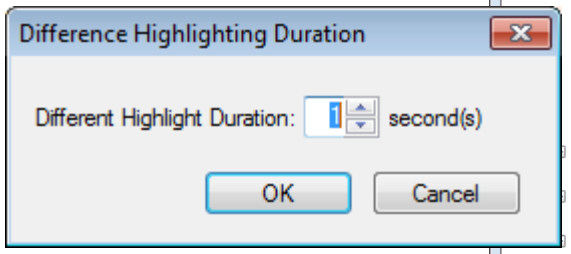


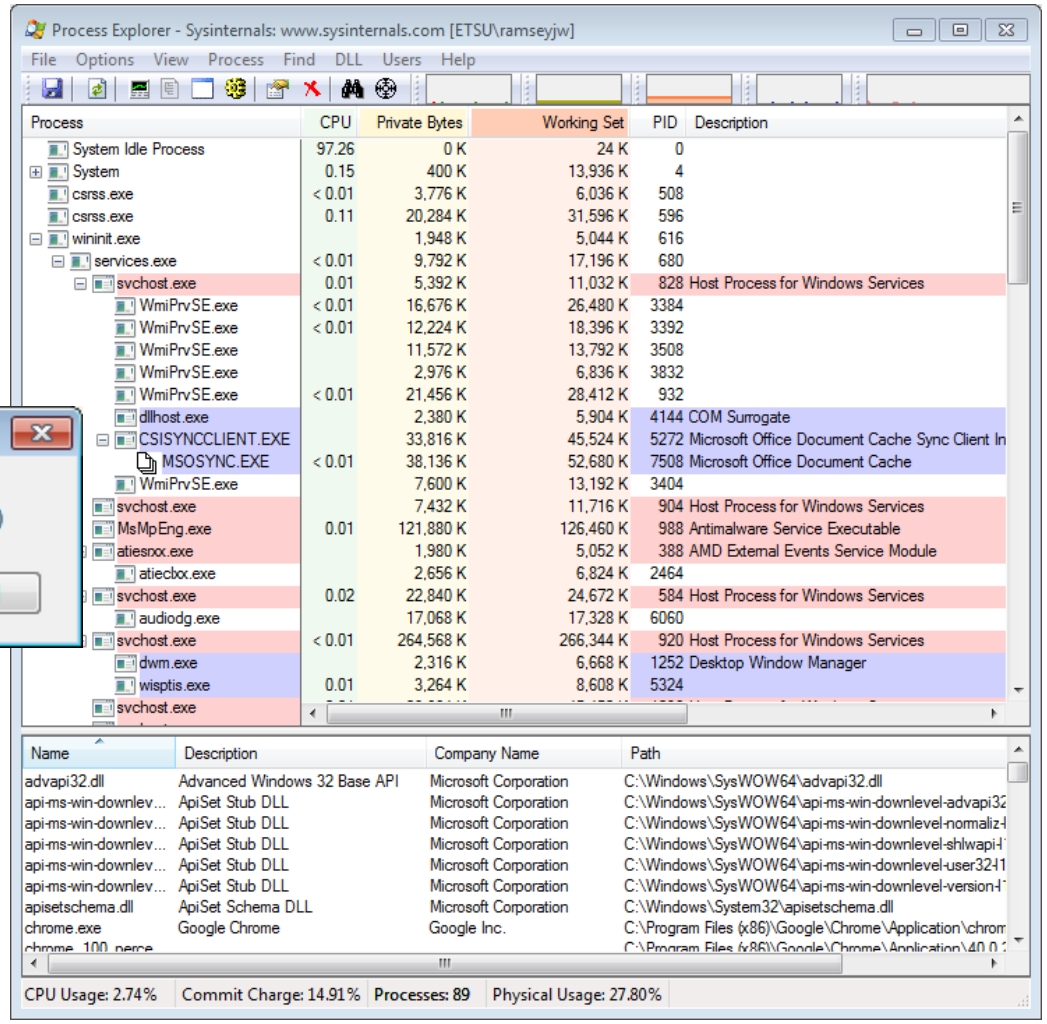
Process Explorer

Highlighting Duration

Configurable: .5-9 sec

Default: 1 sec



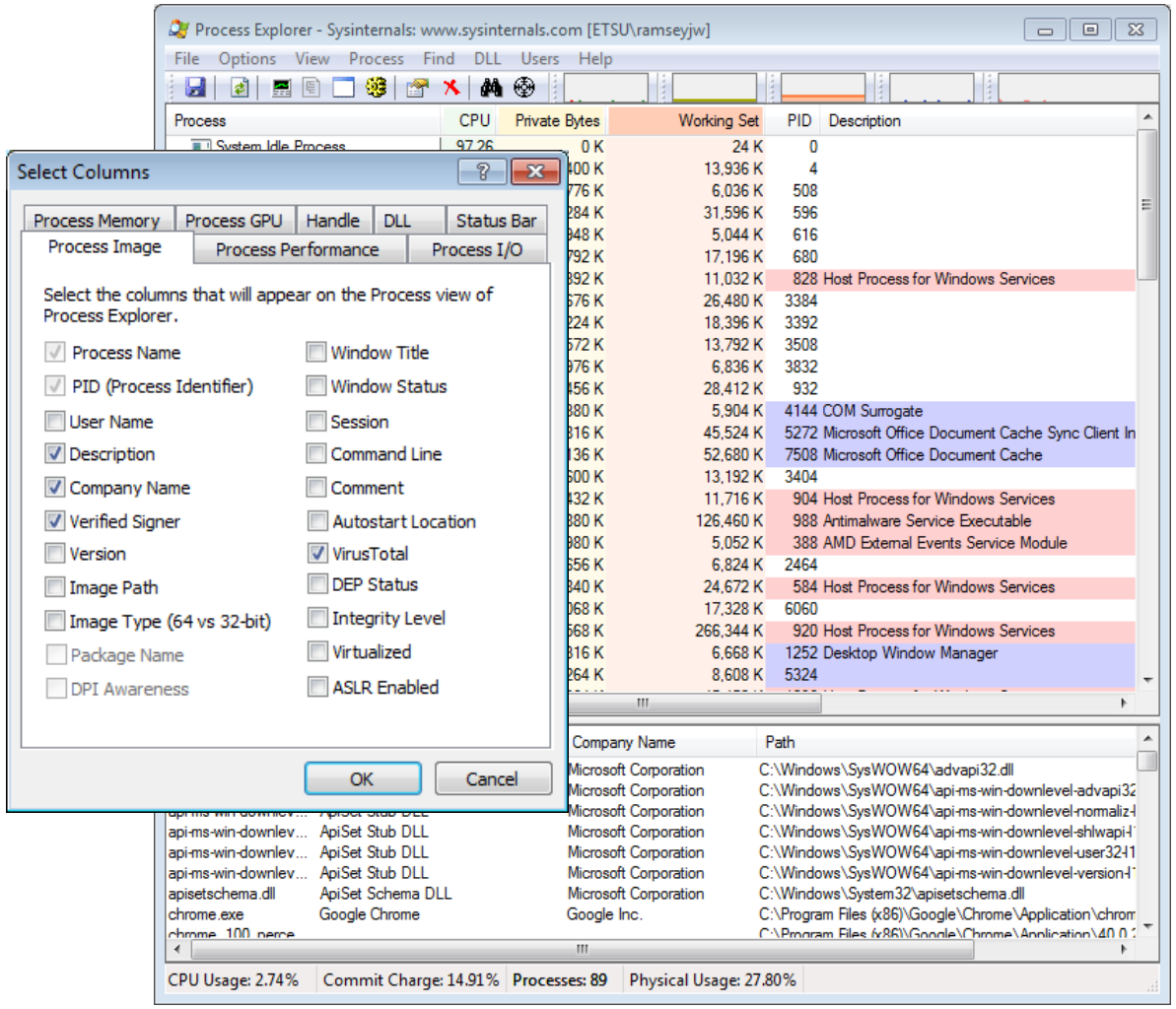


Access: Options->Difference Highlighting Duration

Process Explorer

Highly Configurable

View->Select Columns



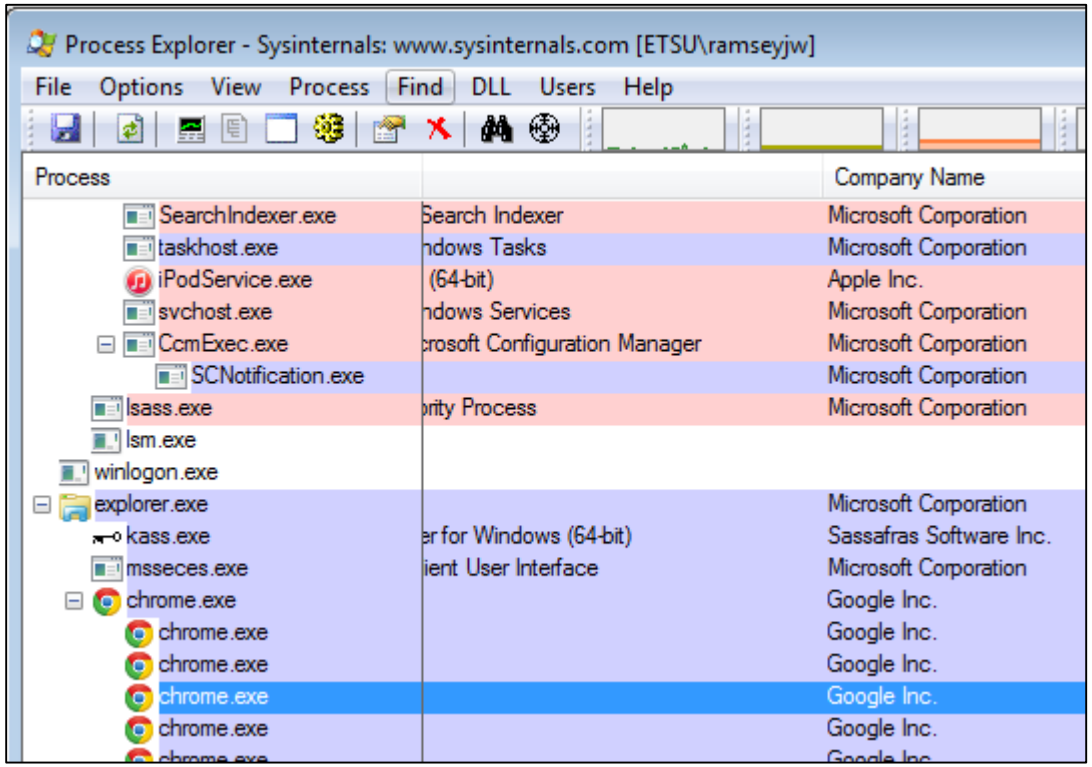
Process Explorer

Company Name & Description

Run as Administrator

Can be helpful in identifying malware

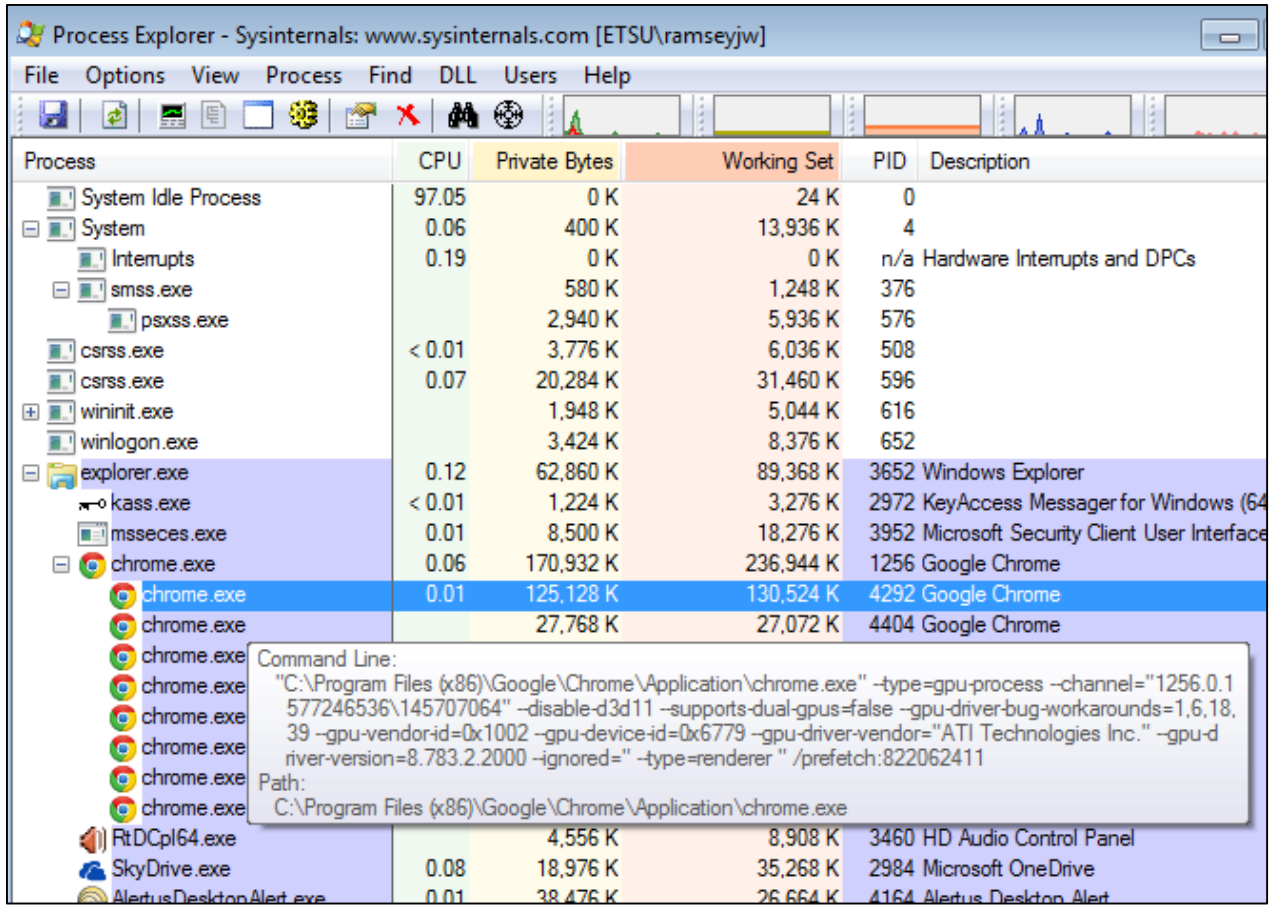
Often not signed



Process Explorer

Tool-tip information

Hovering the cursor over a process provides command-line and path information



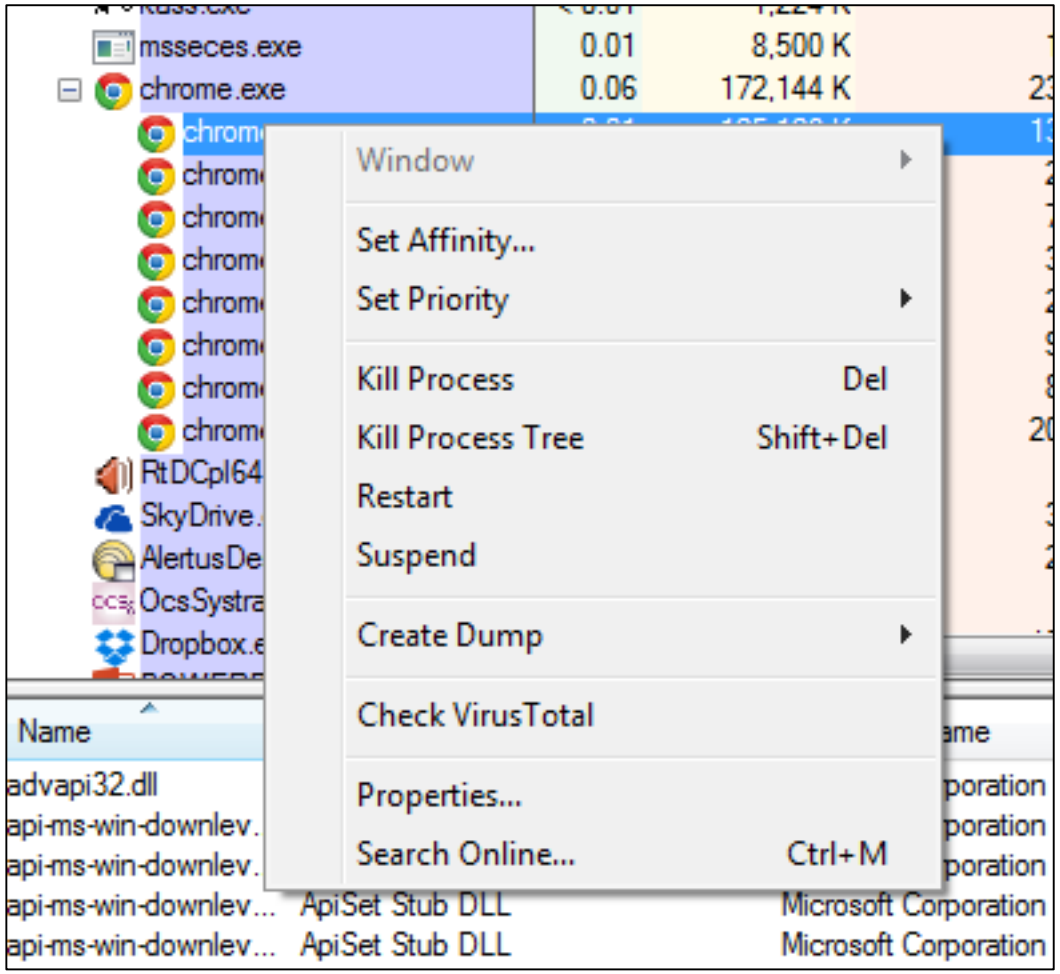
Process Explorer

Process Actions

Right-click on a process

Pop-up window with collection of actions displays

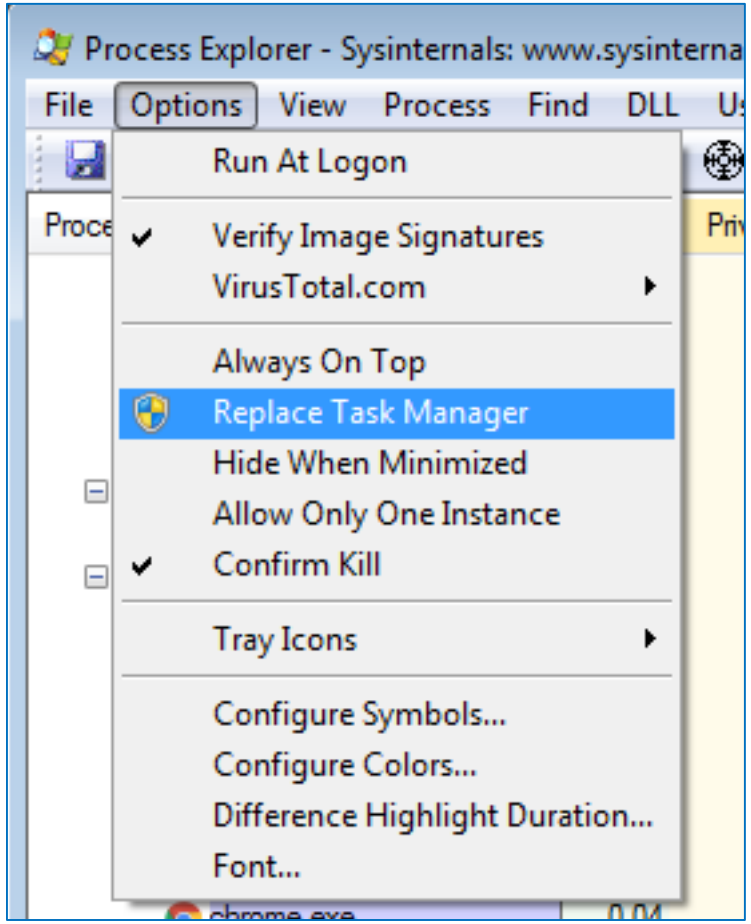
Can also select the process and click on the Process menu



Process Explorer

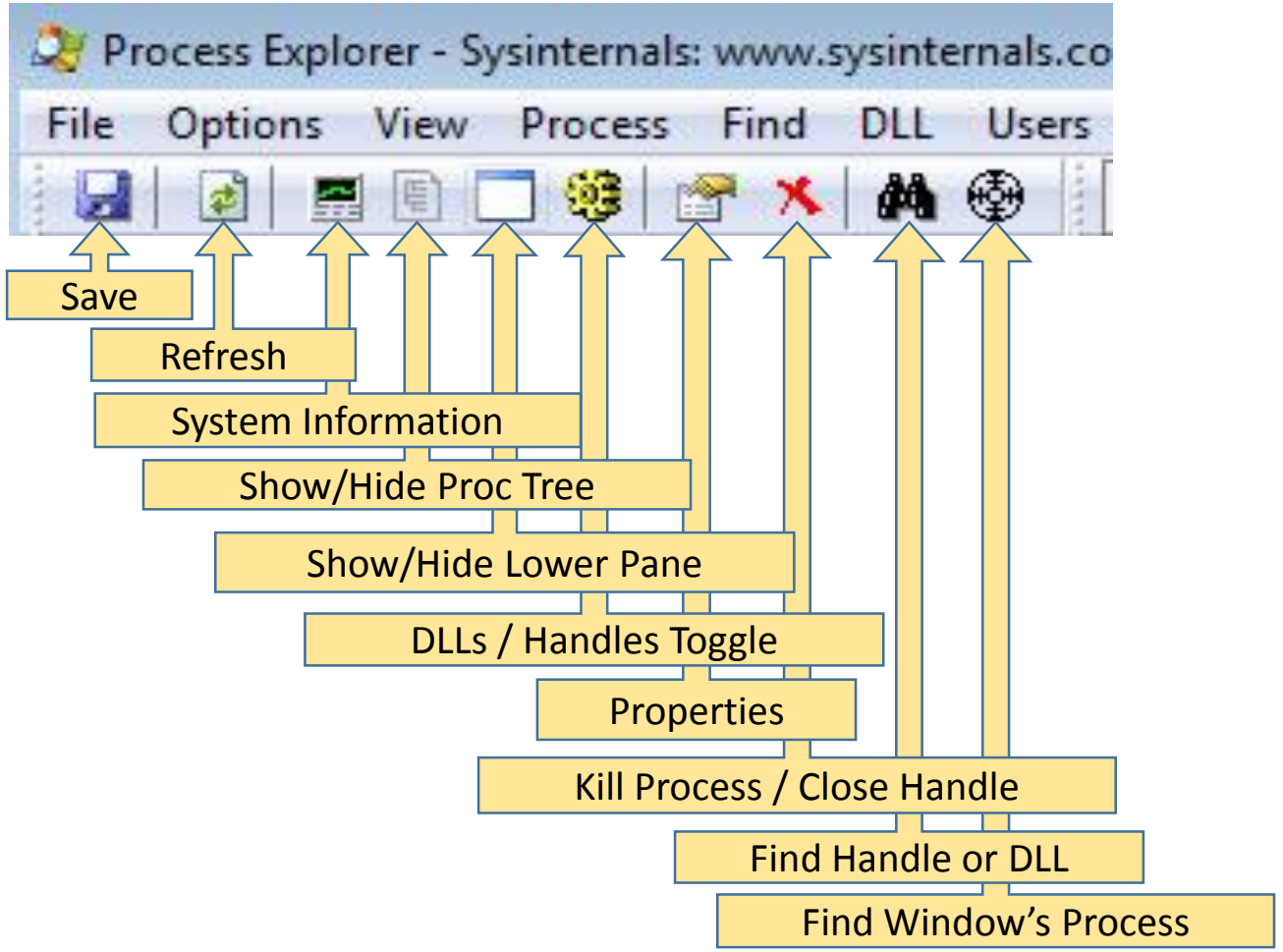
Replace Task Manager

Options->Replace Task Manager



Process Explorer

Toolbar Buttons



Other Sysinternals Tools Related to Processes

Process Monitor

PsTools

VMMap

ProcDump

TCPView

RAMMap

(We may explore some or all of these as we move through the semester. I hope.)

For Lab (2/9/2016)

Read http://www.howtogeek.com/school/sysinternals-pro/lesson2/

(The first page, at least)

Next Class (2/11/2015)

Microsoft Active Directory

(if we have time) Linux management via web console

Questions?

Copyrights

Presentation prepared by and copyright of John Ramsey, East Tennessee State University, Department of Computing . (ramseyjw@etsu.edu)