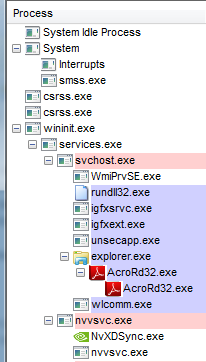
Każdy program w systemie Windows jest reprezentowany przez co najmniej jeden proces. Wiedza o tym, czym są procesy, jak je przeglądać i jak zdobywać o nich szczegółowe informacje, może uchronić przed przykrymi konsekwencjami niestabilności jakiegoś programu.

**Czym jest proces?**

W uproszczeniu można przyjąć, że proces to uruchomiony program. Ważne jest tu słowo „uruchomiony”. Każda aplikacja, czy usługa w systemie Windows działa pod postacią procesu. Proces jest obiektem zawierającym dane i parametry niezbędne do funkcjonowania konkretnego programu, ale nie jest programem jako takim. Pod pojęciem programu należy w takim przypadku rozumieć kod wykonywalny i zestaw danych, czyli pliki zapisane na dysku (lub innym nośniku), proces zaś jest tworzony w momencie uruchomienia jakiegoś programu. Proces zawiera kod uruchomionej aplikacji oraz struktury danych niezbędne do jej funkcjonowania: stos, wskaźnik instrukcji, a także informacje o przyporządkowanych jej zasobach (takie, jak: czas procesora, pamięć, dostęp do urządzeń wejścia-wyjścia).

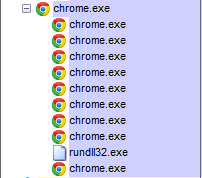
Jednak uruchomienie jednego programu nie jest równoznaczne z utworzeniem przez system jednego procesu. Mnóstwo współczesnych, rozbudowanych narzędzi składa się z wielu rozdzielonych modułów, z których każdy może być wykonywany w postaci odrębnego procesu. Dobrymi przykładami są przeglądarki Internet Explorer 9 i Google Chrome. W obu poszczególne karty i okna ze stronami WWW są widoczne w systemie jako odrębne procesy.

Najczęściej zbiór procesów danego programu ma strukturę hierarchiczną: procesowi nadrzędnemu, tworzonemu w momencie uruchomienia programu, towarzyszą procesy potomne, tworzone w zależności od wywoływanych funkcji. Również sam system operacyjny, uruchamiając kolejno ważne podsystemy, a następnie zależne od nich usługi, buduje w zbiorze procesów hierarchiczną strukturę.



*Fragment hierarchicznej struktury procesów wyświetlony w programie Process Explorer*

Zdarza się również, że aplikacja tworzy w systemie wiele równorzędnych procesów. Takie rozwiązanie często jest stosowane ze względów bezpieczeństwa właśnie we wspomnianych przeglądarkach internetowych: każda karta uruchamiana jest w niezależnym procesie, dzięki czemu wadliwy kod strony powoduje awarię nie całego programu, lecz jedynie procesu, z którym jest powiązany. Po wymuszeniu zamknięcia procesu z niewłaściwie działającą witryną można kontynuować.

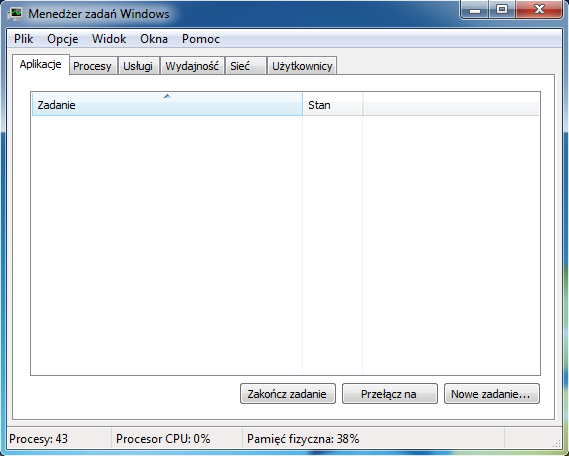


*Fragment drzewa procesów – widoczne procesy Google Chrome. W tej przeglądarce nie tylko każda karta strony WWW jest widoczna jako odrębny proces, ale także każde zainstalowane w programie rozszerzenie. Stąd liczba widocznych procesów może być znaczna nawet przy niewielkiej liczbie otwartych stron*

#### Co w systemie? Menedżer zadań

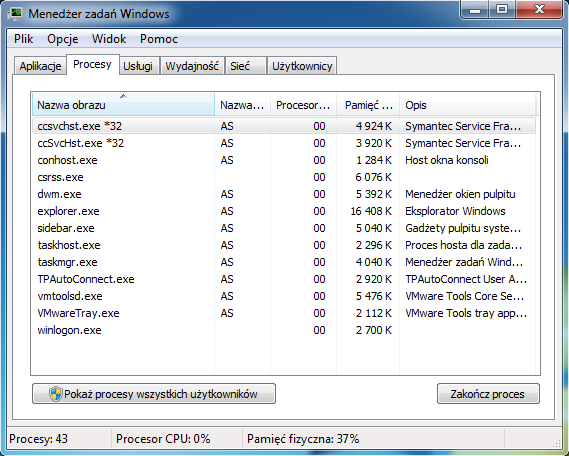
Podstawowym narzędziem umożliwiającym wgląd w zestaw działających procesów jest wbudowany w system Menedżer zadań Windows.

Najprościej uruchomić go skrótem klawiszowym [Ctrl]+[Shift]+[Esc]. Zestaw danych prezentowanych standardowo w jego oknie nie jest szczególnie rozbudowany.



Na kolejnych kartach użytkownik może przejrzeć listę uruchomionych programów, działających procesów, usług, a także dane dotyczące wydajności (mocno okrojone), dane dotyczące sieci (dla każdego interfejsu sieciowego) oraz listę zalogowanych użytkowników. Oprócz tego na pasku stanu w dolnej części okna – bez względu na to, która z kart jest aktywna – program wyświetla liczbę działających procesów, obciążenie procesora oraz odsetek zajętej pamięci. Nie będziemy omawiać wszystkich kart dostępnych w Menedżerze zadań. Skupimy się wyłącznie na tym, jakie informacje o działających procesach systemowych można uzyskać za pomocą tego wbudowanego narzędzia, zarówno w standardowych ustawieniach, jak i po modyfikacjach.

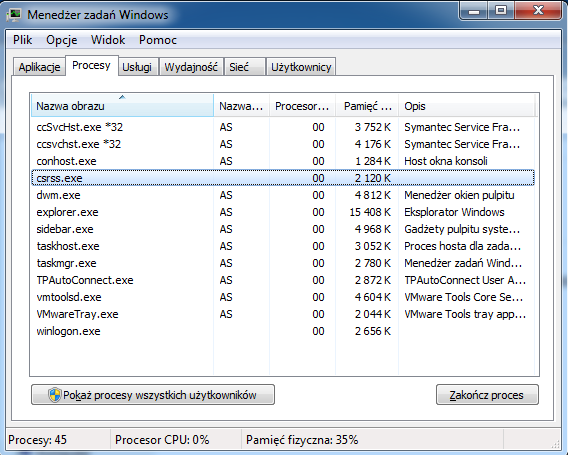
#### Ustawienia standardowe



Lista działających w systemie procesów w standardowej konfiguracji systemu zawiera jedynie pięć parametrów wymienionych w kolejnych kolumnach: ***Nazwa obrazu, Nazwa użytkownika, Procesor CPU, Pamięć (prywatny zestaw roboczy)***oraz***Opis***. Poniżej wyjaśniamy znaczenie tych parametrów.

**Nazwa obrazu -** nazwa procesu. Skąd zatem słowo *obraz*? Wynika to z charakteru samego procesu, który nie jest programem jako takim, lecz reprezentacją (obrazem) jego kodu istniejącą w przyporządkowanej mu przestrzeni adresowej w pamięci. Najczęściej nazwa obrazu jest taka sama jak nazwa pliku wykonywalnego, którego uruchomienie utworzyło nowy proces. Można to wykorzystać do ustalenia pochodzenia aplikacji odpowiadającej za dany proces.

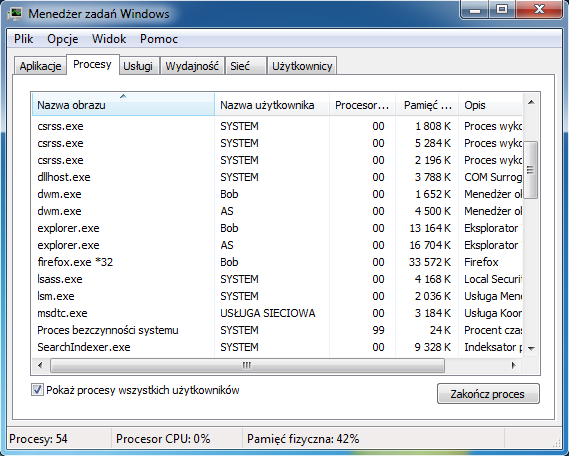
**Nazwa użytkownika - k**ażdy program działający w systemie Windows, a zatem każdy istniejący proces, działa w kontekście konta użytkownika, który go uruchomił. Funkcjonowanie w kontekście konta oznacza, że ma takie same uprawnienia jak właściciel konta. Określenie to obejmuje nie tylko aplikacje świadomie otwarte przez użytkownika, lecz także wszelkie programy uruchomione po zalogowaniu się go w systemie i niewymagające zwiększenia uprawnień (uruchomienie programu wymagającego uprawnień większych od tych, które ma zalogowany użytkownik, objawia się wyświetleniem na ekranie komunikatu wygenerowanego przez mechanizm kontroli konta użytkownika – UAC).



***Menedżer zadań*** standardowo wyświetla tylko procesy zalogowanego użytkownika oraz niektóre procesy systemowe. Procesom, które nie działają w kontekście aktywnego konta, nie towarzyszy żaden wpis w kolumnie ***Nazwa użytkownika***. Na ilustracji widać to na przykładzie *csrss.exe,* jednego z podstawowych procesów Windows.

Takie zachowanie ***Menedżera zadań*** wynika stąd, że w standardowych ustawieniach systemu nawet użytkownik o uprawnieniach administracyjnych nie ma wglądu w parametry procesu uruchomionego w kontekście innego konta niż jego własne.

Oczywiście, ***Menedżer zadań*** umożliwia wyświetlenie wszystkich procesów należących do wszystkich użytkowników zalogowanych w systemie, wystarczy kliknąć przycisk ***Pokaż procesy wszystkich użytkowników***, a następnie zatwierdzić wybór przy użyciu uprawnień administracyjnych.



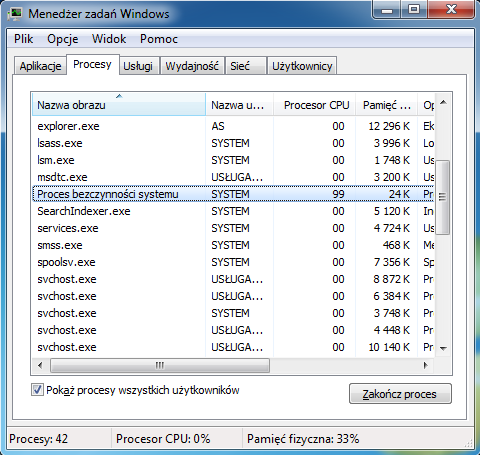
Lista procesów wyświetlona po wykonaniu powyższych działań będzie znacznie dłuższa (ich łączna liczba będzie pokazywana na pasku stanu, przy dolnej krawędzi okna), a w kolumnie ***Nazwa użytkownika*** zostaną wyświetlone nazwy kont wykorzystywanych przez poszczególne procesy. Oprócz procesów powiązanych z kontami użytkowników zalogowanych widoczne będą procesy działające w kontekście kont specjalnych. Do tych kont należą: ***SYSTEM, USŁUGA SIECIOWA*** oraz ***USŁUGA LOKALNA***. Procesy wykonywane w kontekście tych kont są niezależne od użytkownika.

Największe uprawnienia ma proces uruchomiony w kontekście konta ***SYSTEM*** (pełna nazwa tego konta to ***System lokalny***). Daje ono pełny dostęp do wszystkich funkcji systemu, a jeśli komputer pełni również rolę kontrolera domeny, proces uruchomiony na tym koncie może mieć dostęp do całej domeny. Większość podstawowych procesów systemu Windows uruchamiana jest w jego kontekście. Oprócz tego z tych samych uprawnień korzystają np. niektóre moduły programów zabezpieczających dane przed złośliwym oprogramowaniem.

Konta ***USŁUGA LOKALNA*** oraz ***USŁUGA SIECIOWA*** mają znacznie niższy poziom uprawnień, odpowiadający kontom należącym do grupy ***Użytkownicy***. Korzystają z nich procesy, które muszą być uruchamiane niezależnie od użytkownika, ale jednocześnie ze względów bezpieczeństwa nie powinny mieć dostępu do wielu newralgicznych ustawień systemu operacyjnego.

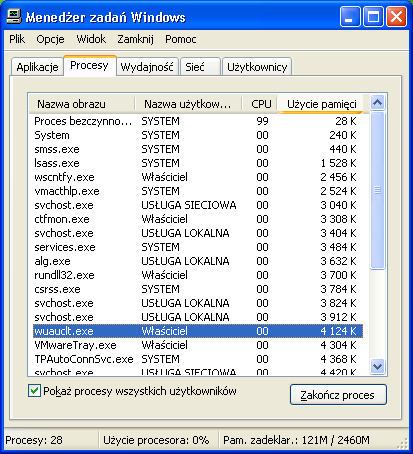
**Procesor i pamięć -** dwie następne kolumny w tabeli procesów wyświetlanej w oknie ***Menedżera zadań*** pokazują wartości, które w bardzo przybliżony sposób pozwalają ocenić, ile zasobów zużywa dany proces. Kolumna ***Procesor CPU*** zawiera po prostu procentowe wskaźniki obciążenia procesora przez dany proces. Wskaźnik ten nie jest zbyt dokładny, bo nie uwzględnia np. podziału pracy między poszczególne rdzenie CPU. Niemniej znaczne obciążenie procesora przez jeden z procesów w czasie, gdy nie zostało uruchomione żadne wymagające obliczeniowo zadanie, może sygnalizować nieprawidłowe działanie jakiegoś programu.

Wyjątkiem od tej reguły jest duże obciążenie w przypadku procesu o nazwie ***Proces bezczynności systemu***.

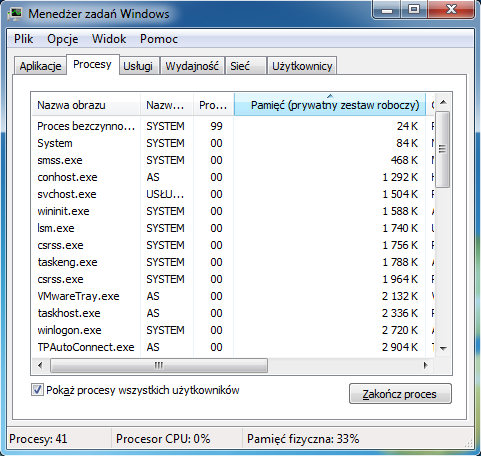


Jego obecność niepokoi wielu mniej doświadczonych użytkowników, którzy widząc „coś” zużywające niemal 100% mocy CPU, sądzą, że przez to komputer działa wolniej, programy nie uruchamiają się itp. (wpisów na wielu forach proszących o poradę, jak „usunąć” ten proces, jest mnóstwo). Tymczasem nie ma najmniejszego powodu do obaw, bo ***Proces bezczynności systemu*** nie jest obrazem jakiegokolwiek programu, lecz systemowym wskaźnikiem ilustrującym, jaki procent mocy obliczeniowej jest niewykorzystany, zatem np. wartość *99* oznacza po prostu, że procesor prawie nic nie robi.

Interpretacja wartości wyświetlanych w kolumnie ***Pamięć*** zależy od wersji Windows. W Windows XP Menedżer zadań wyświetla w niej informacje o wielkości zestawu roboczego pamięci (ang. *working set*).



W Windows Vista Microsoft zmienił standardowe ustawienia.



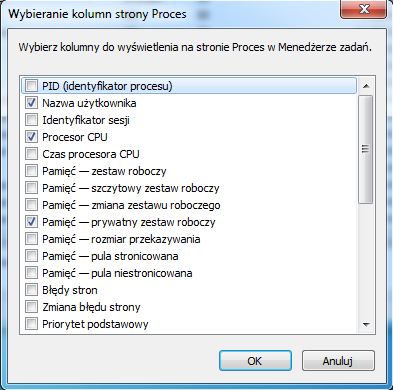
Menedżer zadań w systemach nowszych od Windows XP wyświetla wartość dotyczącą tzw. prywatnego zestawu roboczego pamięci (ang. *privateworking set*).

Różnica pomiędzy tymi wartościami jest dość istotna. Zestaw roboczy (*working set*) obejmuje zarówno pamięć wykorzystywaną wyłącznie przez dany proces, jak i pamięć współdzieloną z innymi procesami, a prywatny zestaw roboczy (*privateworking set*), jest podzbiorem zestawu roboczego i obejmuje niewspółdzieloną pamięć zajmowaną przez dany proces. Zatem szukając odpowiedzi na pytanie o to, ile faktycznie pamięci fizycznej (RAM) zajmuje jakiś program, należy opierać się na prywatnym zestawie roboczym.

**Opis** - zawiera opis poszczególnych procesów. Często wystarczy on do stwierdzenia, z jakimi programami są powiązane, ale niekiedy ich opisy są równie enigmatyczne jak nazwy. Można jednak zmusić ***Menedżera zadań*** do pokazania znacznie większej liczby informacji o nich.

#### Niestandardowe ustawienia

Standardowy zestaw kolumn wyświetlanych w oknie ***Menedżera zadań*** na karcie ***Procesy*** da się znacznie rozbudować. Okno dialogowe pozwalające dodać nowe kolumny można wywołać, wybierając jedną z pozycji z menu ***Widok***. Sam wybór kolumn sprowadza się do zaznaczenia odpowiednich pól.



Jest on o tyle kłopotliwy, że kolumny nie są udokumentowane w pliku pomocy wbudowanym w ***Menedżera zadań***. Poniżej zamieszczamy krótki opis wybranych pozycji przydatnych z punktu widzenia użytkownika, który chce zdiagnozować działające w systemie procesy.

**PID (identyfikator procesu)** – niepowtarzalny identyfikator nadawany każdemu procesowi w momencie jego utworzenia. Parametr ten można wykorzystać np. do sprawdzenia kolejności uruchamiania programów w systemie. Wystarczy posortować listę procesów według PID.

**Identyfikator sesji** – to nic innego jak liczba wskazująca właściciela procesu. Jeżeli w systemie zalogowanych jest wielu użytkowników, to każdy z nich ma inny identyfikator; identyfikatory są przyznawane w kolejności logowania. Identyfikatorem *0* oznaczane są procesy uruchamiane niezależnie od użytkownika (systemowe).

**Czas procesora CPU** – wskaźnik pokazujący całkowity czas użycia procesora (w sekundach) przez dany proces od momentu jego powstania.

**Pamięć - szczytowy zestaw roboczy** – wyświetla maksymalny zestaw roboczy pamięci, jaki został przydzielony danemu procesowi od momentu uruchomienia go w danej sesji. Wszystkie wskaźniki Menedżera zadań dotyczące pamięci wyświetlają wartości w kilobajtach.

**Zmiana zestawu roboczego (pamięć)** – wartość pokazująca, o ile zmienił się rozmiar zestawu roboczego pamięci danego procesu od ostatniej aktualizacji (w domyślnych ustawieniach ***Menedżera zadań*** – co sekundę). To dość istotny wskaźnik, pozwalający łatwo zdiagnozować aplikacje z tzw. wyciekiem pamięci (ang. *memoryleak*) – poważnym błędem w kodzie programu, który w trakcie działania nie umie zwolnić niepotrzebnej już pamięci, tylko wciąż rezerwuje nowe zasoby. Stale widoczna wartość dodatnia może sugerować wadliwie działającą aplikację, która prędzej czy później może spowodować niestabilność systemu na skutek zajęcia wszystkich dostępnych zasobów w RAM-ie.

**Pamięć - rozmiar przekazywania** – ilość pamięci wirtualnej zastrzeżonej do użycia przez dany proces.

**Pamięć - pula stronicowana** – wskaźnik ten pokazuje, jaka część pamięci przydzielonej danemu procesowi może zostać zapisana na innym nośniku, co w praktyce oznacza, ile pamięci procesu może być przeniesione np. do pliku wymiany.

**Pamięć - pula niestronicowana** – część pamięci przydzielonej danemu procesowi, która nie może zostać zapisana na innym nośniku. Wyświetlana wartość oznacza zatem pamięć, która nie może zostać zwolniona przez alokowanie do pliku wymiany (zwanego też plikiem stronicowania).

**Błędy stron** – licznik zdarzeń polegających na tym, że dany proces odwołuje się do strony w pamięci, która nie istnieje w przydzielonym procesowi zestawie pamięci; następuje wtedy przerwanie sprzętowe i właściwa strona zostaje odczytana z pliku wymiany. W terminologii informatycznej błędy stron również są interpretowane jako poważne awarie sprzętu, jednak w ***Menedżerze zadań*** należy ten wskaźnik traktować jedynie jako licznik odwołań do pamięci wirtualnej wykonanych przez dany proces.

**Zmiana błędu strony** – parametr pokazujący, jak zmienia się liczba odwołań do pamięci wirtualnej w danym procesie. Jeżeli wartość ta jest stale większa od zera albo znacząco rośnie, może to oznaczać, że po prostu systemowi lub działającym w nim aplikacjom zaczyna brakować RAM-u. Wskaźnik ten jest standardowo aktualizowany co sekundę, zatem np. wartość *12* w tej kolumnie oznacza, że dany proces w ciągu ostatniej sekundy odwołał się do pamięci wirtualnej 12 razy.

**Priorytet podstawowy** – wskaźnik informujący, jaki priorytet dostępu do procesora ma dany proces. Wątki procesu o najwyższym priorytecie są wykonywane w pierwszej kolejności.

**Wątki** – informuje o liczbie wątków uruchomionych w danym procesie; wątek jest wydzielonym zestawem instrukcji programu wykonywanym współbieżnie z innymi wątkami w ramach tego samego procesu. Każdy proces składa się przynajmniej z jednego wątku, w praktyce w każdej aplikacji Windows jest ich zawsze więcej (jednowątkowe były aplikacje pisane dla „systemu” DOS).

**Odczyty We/Wy** – wskaźnik zliczający wygenerowaną przez dany proces liczbę operacji odczytu wejścia-wyjścia (czyli odczytu pliku, sieci lub urządzenia; wskaźnik ten pomija odczyty kierowane do obiektu wejścia konsoli).

**Zapisy We/Wy** – jak wyżej, z tą różnicą, że dotyczy operacji zapisu.

**Inne We/Wy** – jak wyżej, ale dotyczy operacji innych niż odczyt i zapis (np. funkcji kontrolnych).

**Odczyty We/Wy w bajtach** – pokazuje, ile bajtów zostało odczytanych w operacjach wejścia-wyjścia wygenerowanych przez dany proces od momentu uruchomienia go. Również w tym przypadku operacje na obiekcie wejścia konsoli nie są liczone.

**Zapisy We/Wy w bajtach** – jak wyżej, ale dotyczy zapisanych bajtów.

**Inne We/Wy w bajtach** – jak wyżej, ale dotyczy bajtów przesłanych w operacjach wejścia-wyjścia innych niż odczyt i zapis.

**Nazwa ścieżki obrazu** – jest to ścieżka wyświetlająca dokładną lokalizację pliku wykonywalnego odpowiedzialnego za powstanie danego procesu.

**Wiersz polecenia** – parametr podobny do poprzedniego, ale tym razem oprócz samej ścieżki widoczna jest informacja o tym, z jakimi parametrami dany program został uruchomiony.

**Wirtualizacja funkcji Kontrola konta użytkownika** – wskaźnik pokazujący, czy dla danego procesu funkcja wirtualizacji UAC (ang. *User Account Control*) jest włączona, wyłączona czy też w ogóle niedozwolona.

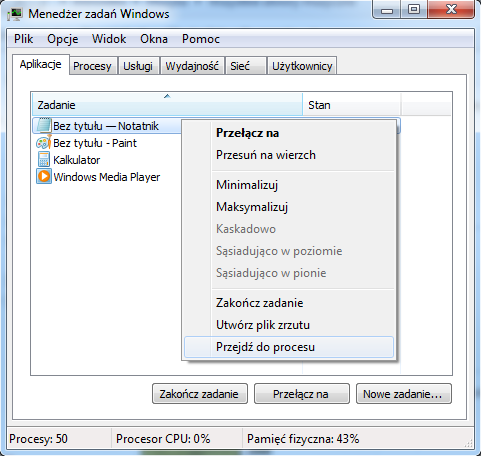
**Zapobieganie wykonywaniu danych** – wskaźnik pokazujący, czy dla danego procesu włączona jest funkcja zapobiegania wykonywaniu danych (DEP – *Data ExecutionPrevetion*; jest to zabezpieczenie chroniące przed wykonaniem nieautoryzowanego – w domyśle: złośliwego – kodu z segmentu danych, zapewnia ochronę przed atakami wykorzystującymi przepełnienie bufora).

Jak widać, zestaw informacji na temat danego procesu można znacznie rozszerzyć w stosunku do skromnego zbioru wyświetlanego w standardowej konfiguracji ***Menedżera zadań***.

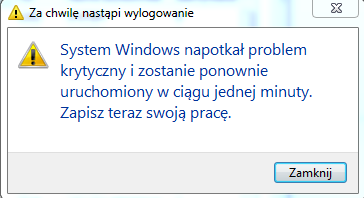
#### Zamykanie procesów

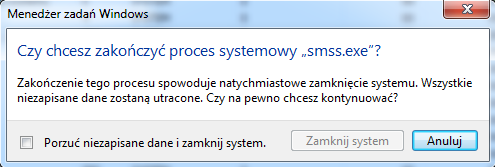
***Menedżer zadań*** nie tylko umie wyświetlać informacje o poszczególnych procesach uruchomionych w systemie, ale także pozwala wpływać na ich działanie. To narzędzie jest najczęściej wykorzystywane do awaryjnego zamykania procesów wadliwie działających aplikacji.

W sytuacji gdy jakiś program przestaje poprawnie działać, co najczęściej objawia się brakiem możliwości jakiekolwiek interakcji z nim, nie ma innego wyjścia, jak wymusić zamknięcie należącego do niego procesu. Nie zawsze jednak nazwa procesu jest na tyle czytelna, by można było rozpoznać, z jakim programem jest on powiązany. Oczywiście, można posłużyć się informacjami wyświetlanymi na karcie ***Procesy*** (np. nazwą ścieżki obrazu czy opisem), ale istnieje prostszy sposób.



Należy wywołać ***Menedżera zadań*** i uaktywnić kartę ***Aplikacje***, a następnie kliknąć prawym przyciskiem nazwę programu, który przestał działać, i z menu kontekstowego wybrać pozycję ***Przejdź do procesu***. Menedżer zadań automatycznie uaktywni kartę ***Procesy*** i zaznaczy odpowiednią pozycję. Wystarczy wtedy kliknąć przycisk ***Zakończ proces***, by usunąć źródło problemu. Metoda ta zadziała jednak tylko w przypadku procesów uruchomionych przez użytkownika oraz tych, które działają w kontekście kont o mniejszych uprawnieniach (takich, jak ***USŁUGA LOKALNA*** i ***USŁUGA SIECIOWA***). Próba zamknięcia procesu systemowego może skutkować odmową dostępu do danego procesu lub wyświetleniem jednego z komunikatów pokazanych na poniższych ilustracjach.





Tym samym łatwo zauważyć, że kończenie działania niektórych procesów po to, aby odzyskać dostęp do systemu, nie ma najmniejszego sensu, ponieważ powoduje to automatyczny restart komputera.

#### Czego nie zamykać?

Oto krótka lista procesów systemowych Windows 7, które nigdy nie powinny być wyłączane przez wymuszenie zamknięcia.

**Csrss.exe** – jeden z podstawowych komponentów systemów Windows (***client/server run-timesubsystem***), odpowiedzialny m.in. za okna konsoli czy tworzenie i usuwanie wątków. Jest to część podsystemu Win32 działająca w tzw. trybie użytkownika (ang. ***usermode***; tryb użytkownika, zwany też przestrzenią użytkownika, oznacza przestrzeń adresową, do której dostęp mają zarówno komponenty jądra systemu, jak i aplikacje). Bez względu na liczbę kopii csrss.exe w systemie powinien on się znajdować wewnątrz folderu systemowego (***C:\Windows\system32***). Uwaga: jeżeli lokalizacja procesu jest inna, to bardzo prawdopodobne, że masz do czynienia ze złośliwym kodem podszywającym się pod proces systemowy.

**Lsass.exe** – jest to proces lokalnego serwera uwierzytelniania zabezpieczeń, odpowiada za uwierzytelnianie użytkowników w usłudze ***Winlogon***. Proces ten jest niezbędny m.in. dlatego, że generuje tzw. token dostępu użytkownika, wykorzystywany do uruchomienia początkowej powłoki. Wszystkie dalsze procesy użytkownika dziedziczą ten token. Prawdziwy proces *lsass.exe* rezyduje w folderze *\****Windows****\****System32****\*; inna lokalizacja sygnalizuje obecność złośliwego kodu podszywającego się pod proces systemowy.

**Services.exe** – Service Control Manager, proces zarządzający działaniem usług systemowych Windows, odpowiada za uruchamianie i zatrzymywanie usług systemowych oraz komunikowanie się z nimi. Właściwa lokalizacja to folder***\Windows\System32****\*; jakakolwiek inna najprawdopodobniej oznacza złośliwe oprogramowanie.

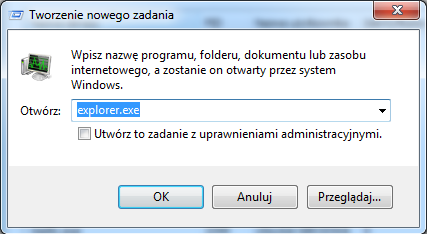
**Smss.exe** – podsystem menedżera sesji (ang. ***session manager***); proces uruchamiający sesję użytkownika, inicjowany przez wątek systemowy,  odpowiedzialny m.in. za uruchomienie procesu logowania (***Winlogon.exe***) oraz procesu ***csrss.exe***, a także ustawienie zmiennych systemowych. Prawdziwy *smss.exe* zlokalizowany jest w folderze***\Windows\System32***; jakakolwiek inna lokalizacja może oznaczać obecność szkodliwego kodu.

**Spoolsv.exe** – zarządza zadaniami drukowania/faksowania. Dzięki niemu możliwe jest m.in. drukowanie w tle. Prawidłowa lokalizacja to***\Windows\System32\;*** jakakolwiek inna oznacza złośliwy kod.

**Svchost.exe** – uniwersalny proces pełniący rolę gospodarza innych procesów uruchamianych w systemie. Może występować na liście ***Menedżera zadań*** wielokrotnie i jest to normalne. Wszystkie kopie prawidłowego ***svchost.exe*** powinny mieć tę samą ścieżkę obrazu: ***C:\Windows\System32\svchost.exe***; każda inna oznacza złośliwe oprogramowanie.

**System** – proces systemowy związany z wątkami kodu uruchamianymi w trybie jądra.

Nie należy również wyłączać procesu o nazwie ***explorer.exe***, który reprezentuje interaktywną, graficzną powłokę systemu Windows. Zamknięcie go nie skutkuje co prawda natychmiastowym restartem komputera, ale powoduje zniknięcie interfejsu użytkownika: ikon, paska zadań, menu ***Start*** itp. Niestety, czasem ten proces trzeba zamknąć, by odzyskać dostęp do działających w systemie programów. Może się zdarzyć sytuacja, w której jakiś program próbuje wykonać operację na plikach, odwołuje się do powłoki (***explorer.exe***) i zawiesza. W takim przypadku może się okazać konieczne nie tylko zamknięcie procesu danej aplikacji, ale również procesu powłoki (***explorer.exe***). Bardzo ważne jest, by nie zamykać po tym okna Menedżera zadań.

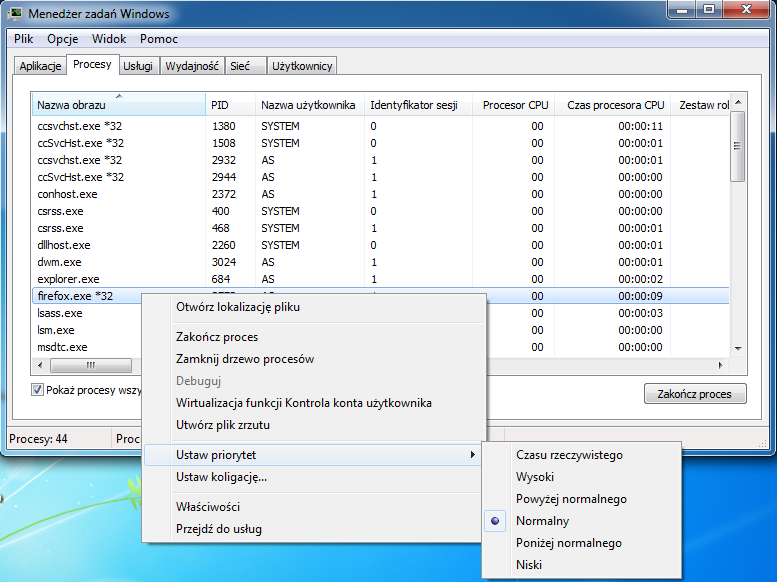


Samo zamknięcie procesu ***explorer.exe*** rozwiąże problem z zawieszoną powłoką, ale nie spowoduje odblokowania dostępu do pozostałych programów uruchomionych w systemie, bo choć działają, nie można się do nich dostać poprzez interfejs graficzny ze względu na brak paska zadań, menu ***Start*** i jakichkolwiek okien (poza oknem ***Menedżera zadań*** – o ile nie został wcześniej zamknięty). ***Menedżer zadań*** umożliwia jednak powtórne uruchomienie graficznej powłoki: wystarczy z menu ***Plik*** wybrać opcję uruchomienia nowego zadania, a następnie w wyświetlonym oknie dialogowym, widocznym na powyższej ilustracji, wprowadzić nazwę pliku uruchamiającego proces ***explorer.exe***. Powłoka graficzna zostanie ponownie załadowana i będzie możliwe np. zapisanie zmian w otwartych aplikacjach.

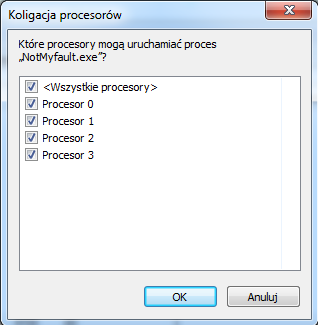
W przypadku procesów systemowych bardzo ważne jest, aby zwracać uwagę na lokalizację powiązanych z nimi plików wykonywalnych. Jeżeli jest inna niż wskazana w opisie (najczęściej jest to systemowy folder ***C:\Windows\System32*** – oczywiście przy założeniu, że system operacyjny jest zainstalowany na dysku oznaczonym literą ***C:*** w folderze***\Windows***), to mamy do czynienia ze złośliwym oprogramowaniem.

#### Manipulowanie procesami

Zamykanie procesów wadliwie działających programów czy też uruchamianie zadań, to nie wszystko, co można zrobić z poziomu ***Menedżera zadań.***



Dostęp do wszystkich działań, które można wykonać na wybranych procesach, zapewnia menu kontekstowe danego obiektu. Z ciekawszych opcji warto wymienić możliwość otwarcia lokalizacji pliku wykonywalnego w oknie ***Eksploratora Windows***, ustawienia priorytetu dostępu do procesora oraz ustawienia koligacji, czyli określenia, na ilu rdzeniach procesora mogą być wykonywane wątki danego procesu.



Standardowo wszystkie wątki każdego procesu uruchomionego w systemie mogą korzystać ze wszystkich dostępnych rdzeni CPU, zatem ustawienie koligacji nie jest funkcją optymalizacyjną systemu, to raczej funkcja diagnostyczna, pozwalająca np. ograniczyć wykonywanie wątków wybranego procesu do jednego rdzenia.

W systemie Windows 7 każda aplikacja, nawet taka, której kod nie został zoptymalizowany pod kątem działania na wielu rdzeniach, wykorzystuje wszystkie dostępne zasoby CPU. Oczywiście, takie systemowe rozłożenie wątków na wiele rdzeni nie zapewnia wydajności kodu zoptymalizowanego pod kątem wykonywania na wielu rdzeniach, ale i tak jest lepszym rozwiązaniem niż ograniczenie działania programu do jednego rdzenia. Niestety, może się zdarzyć, że jakiś program nie podda się systemowemu mechanizmowi równoważenia obciążenia wielordzeniowego procesora i po prostu przestanie działać. Jeżeli problem występuje regularnie, to warto spróbować za pomocą ustawień koligacji ograniczyć wykorzystanie rdzeni CPU.

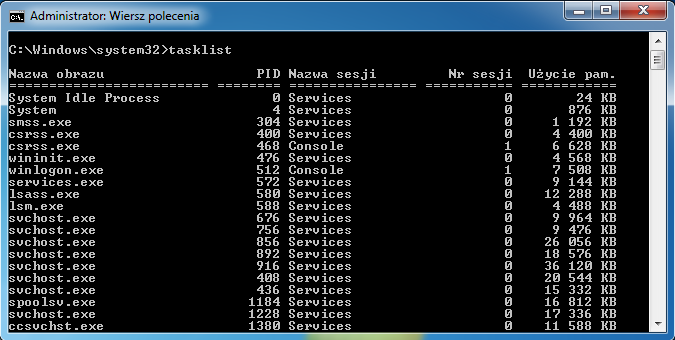
**Narzędzia tekstowe – Wiersz polecenia**

Menedżer zadań jest narzędziem wygodnym w obsłudze, ale ogranicza go przede wszystkim to, że ustawienia, które udostępnia, dotyczą wyłącznie komputera, na którym program jest uruchomiony. Osobom zarządzającym nawet niewielką grupą pecetów perspektywa przesiadania się z jednego na inny po to, aby sprawdzić, jakie procesy działają na każdym z nich, z pewnością nie wydałaby się atrakcyjna.

**Wiersz polecenia**

Wiersz polecenia towarzyszy systemom Microsoftu od zawsze. Konsola ***cmd.exe*** wraz z towarzyszącym jej zestawem poleceń jest cały czas wykorzystywana przez wielu administratorów i użytkowników Windows. Wiersza polecenia można użyć również do zarządzania procesami, zarówno w trybie interaktywnym (czyli uruchamiasz konsolę, wpisujesz polecenie i wykonujesz je), jak i w trybie wsadowym (tworzysz plik ***.bat***zawierający zestaw poleceń dotyczących procesów).

Do zarządzania procesami z poziomu ***Wiersza polecenia*** warto stosować dwa polecenia: ***tasklist*** oraz ***taskkill***. Pierwsze wyświetla listę uruchomionych w systemie aplikacji oraz skojarzonych z nimi procesów, drugie zaś pozwala awaryjnie zakończyć jeden lub kilka procesów. Oba można stosować zarówno do systemu lokalnego, jak i zdalnego.



Wywołanie polecenia ***tasklist*** bez żadnych parametrów powoduje wyświetlenie na ekranie listy uruchomionych procesów. Zestawienie to wygląda nieco inaczej niż w ***Menedżerze zadań***. Przede wszystkim oprócz nazwy pokazywany jest identyfikator procesu (***PID***) i według niego jest sortowana lista. Identyfikator sesji jest tutaj uzupełniony o nazwę sesji (sesja użytkownika nazywa się ***Console***), a wartości w kolumnie ***Użycie pamięci*** dotyczą nie prywatnego zestawu roboczego, jak w przypadku Menedżera zadań Windows Vista/7, lecz ogólnie całego zestawu roboczego danego procesu (tak jak w Menedżerze zadań Windows XP).

Aby otrzymać listę działających procesów w pliku tekstowym, wystarczy przekierować wyjście konsoli do pliku.

http://pclab.pl/zdjecia/artykuly/serafinowicz/2012Procesy/tasks02.png

W efekcie na dysku, w miejscu wskazywanym przez pełną ścieżkę dostępu zawartą w poleceniu, powstanie plik zawierający spis działających procesów lokalnego komputera.

***tasklist***, podobnie jak wiele innych, można wykonać z licznymi przełącznikami. Ich krótki opis zostanie wyświetlony po wywołaniu polecenia ***tasklist /?.***Warto się z nimi zapoznać, bo znacznie rozszerzają one funkcjonalność tego narzędzia. Oto kilka użytecznych przykładów:

* ***tasklist /S testowy /U Kowalski /FI "STATUS eq NOT RESPONDING"***  
  Polecenie to wyszuka wszystkie nieodpowiadające (ang. ***not responding***) procesy na komputerze o nazwie ***testowy*** uruchomione w kontekście konta użytkownika o nazwie ***Kowalski***. Wynik oczywiście można przekierować do pliku.
* ***tasklist /V /FO CSV > C:\procesy.csv***  
  To polecenie utworzy na dysku C: (w katalogu głównym) plik w formacie CSV (ang. ***commaseparatedvalues*** – 'wartości oddzielone przecinkami') zawierający listę procesów uzupełnioną o dodatkowe informacje (przełącznik */V*) na temat każdego z nich. Taki plik po zaimportowaniu np. do Excela jest przejrzystym zbiorem danych o procesach.
* ***takslist /SVC***Proste i użyteczne polecenie, które wyświetla listę procesów uzupełnioną o informacje na temat usług systemowych działających w ramach wybranego procesu.

Drugim wartym poznania poleceniem jest ***taskkill***. Używa ono podobnych przełączników jak ***tasklist***. Jego największą zaletą jest to, że pozwala ono zakończyć automatycznie wiele procesów jednocześnie. Z drugiej strony taka funkcjonalność może okazać się niebezpieczna w rękach niedoświadczonego użytkownika.

Oto przykłady użycia:

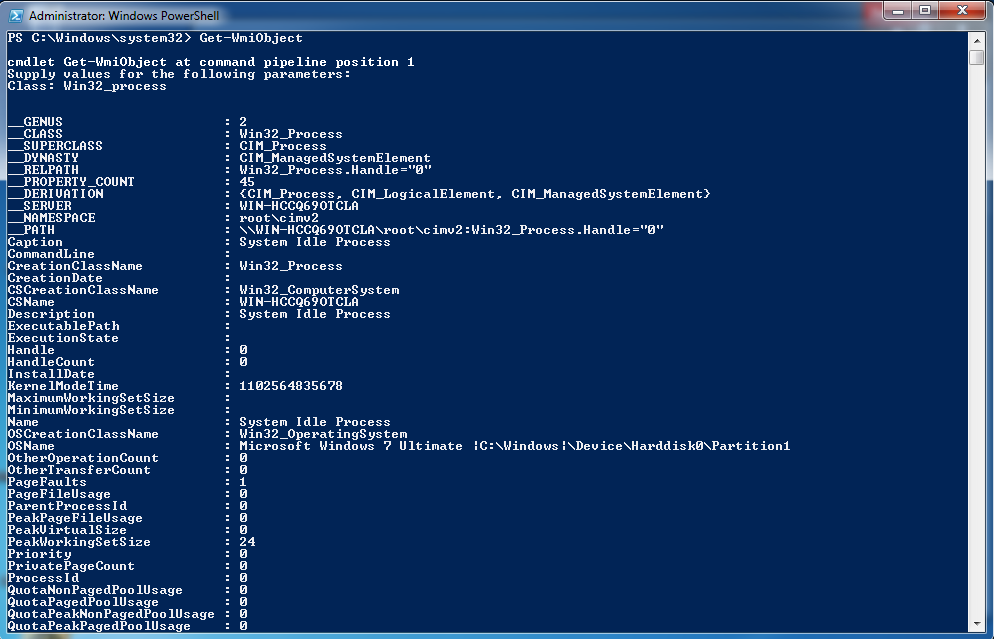
* ***taskkill /IM firefox.exe*** spowoduje tylko zakończenie działania procesu popularnej przeglądarki WWW, ale polecenie:
* ***taskkill /IM f\****spowoduje zamknięcie wszystkich procesów, których nazwa zaczyna się na literę *f*.

Informacje o wszystkich opcjach polecenia ***taskkill***można uzyskać po wywołaniu go z parametrem***/?***.

**PowerShell i klasa win32\_process**

Windows 7 jest standardowo wyposażony w dodatkowy interpreter o nazwie ***Windows PowerShell***. Jest on znacznie bardziej funkcjonalny i rozbudowany niż ***cmd.exe***. ***PowerShell*** jest zintegrowany z ***.NET Framework*** i pozwala wywołać każdą funkcję dostępną w ramach całego API Windows. Oznacza to, że można dzięki niemu wykonać każde programowalne zdarzenie w systemie; gromadzenie informacji o działających w nim procesach nie przekracza jego możliwości. Użytkownicy starszych wersji Windows również mogą go używać, ale muszą najpierw pobrać i zainstalować odpowiedni pakiet.

Za pomocą ***PowerShella*** można korzystać także z poleceń ***Wiersza polecenia***, ale znacznie większe możliwości wydobycia szczegółowych informacji o procesach zapewni wbudowany w Windows 7 podsystem ***Windows Management Instrumentation (WMI)***, a dokładniej: klasa *Win32\_****process***.



Na przykład wywołanie w oknie ***Windows PowerShell*** polecenia:

***Get-WmiObject -class win32\_process > C:\result.txt***

...spowoduje zapisanie do pliku o **nazwie *result.txt*** szczegółowych informacji o działających w systemie procesach.

Oczywiście, informacje zwracane przez polecenie ***Get-WmiObject*** można gromadzić także z innych komputerów dostępnych w sieci, a wyniki filtrować za pomocą polecenia ***Select-Object***. Na przykładpolecenie:

***Get-WmiObject -computernametestowy -class win32\_process | Select-Object name, processID***

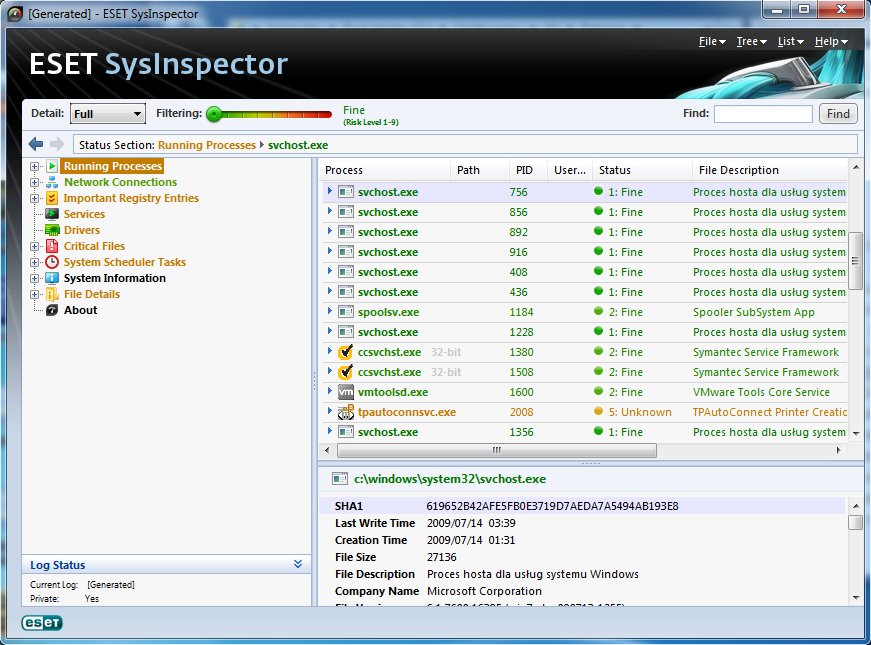
...spowoduje wyświetlenie listy nazw i identyfikatorów (PID) procesów działających na komputerze o nazwie ***testowy***.

**Dodatkowe programy**

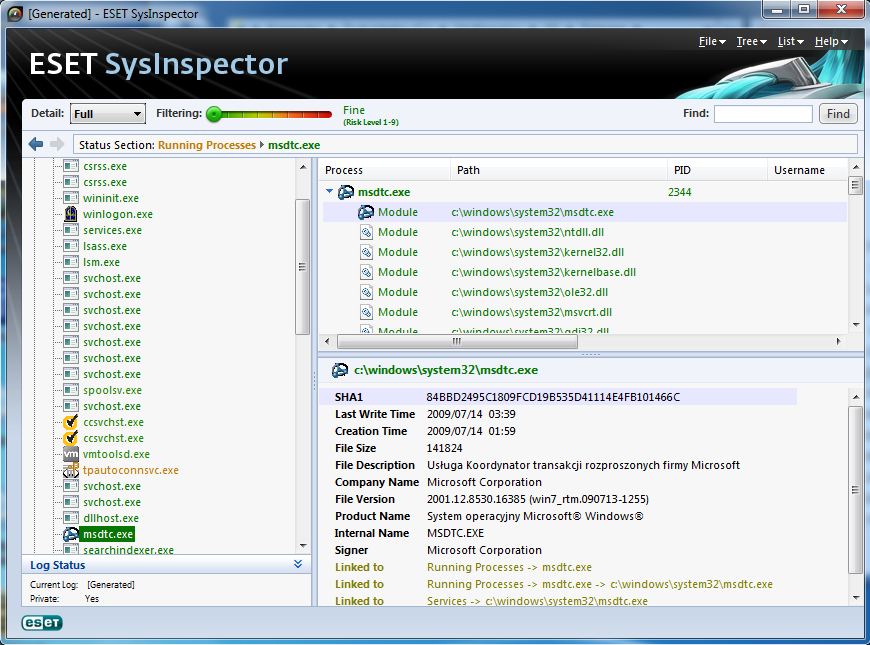
Choć trudno mieć zastrzeżenia co do dokładności i szczegółowości informacji o procesach, które można otrzymać, korzystając z ***PowerShella*** i klasy *Win32\_****process***, to przejrzystość tego rozwiązania jest daleka od doskonałości.

[**ESET SysInspector**](http://pliki.onet.pl/ESET.SysInspector,Informacje.i.diagnostyka,Windows,111490.html)

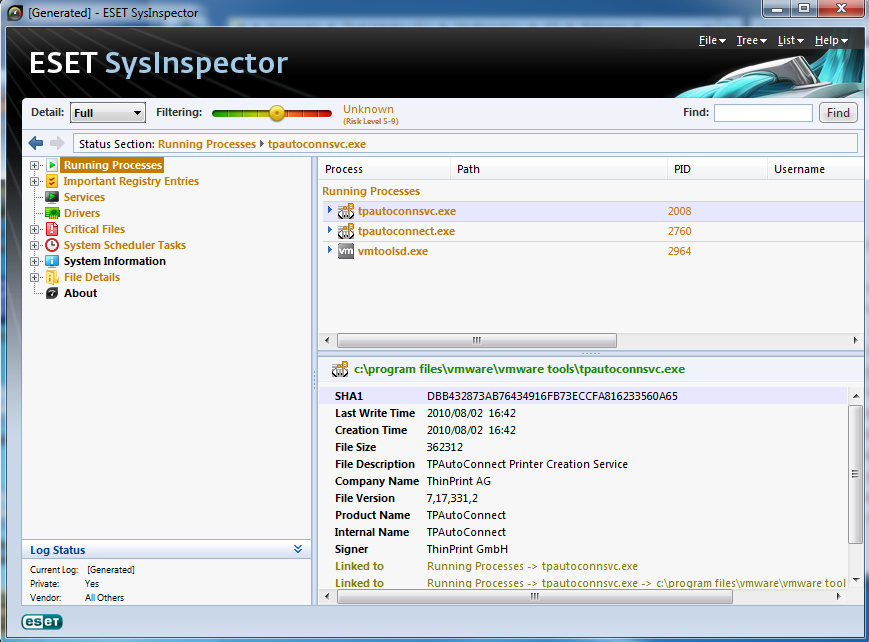
Bezpłatny program opracowany przez renomowanego słowackiego producenta rozwiązań antywirusowych i zabezpieczających. ***SysInspector*** to dość proste w obsłudze, niewymagające instalacji narzędzie diagnostyczne przeznaczone w szczególności do analizy kodu, który w danym momencie jest uruchomiony w systemie. Program nie tylko wyświetla listę uruchomionych procesów, ale też wiąże te informacje z otwartymi połączeniami sieciowymi, wpisami do rejestru systemowego, działającymi usługami, sterownikami, krytycznymi plikami systemu operacyjnego, a także wpisami harmonogramu zadań.



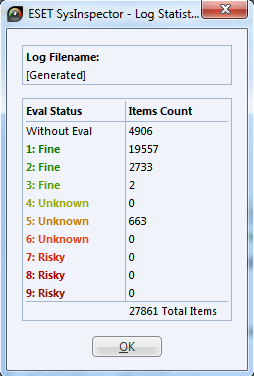
Wszystkie informacje wyświetlane przez ***SysInspectora*** mogą być filtrowane, ale w standardowej konfiguracji program prezentuje pełny obraz tego, co dzieje się w systemie. Zaletą na tle ***Menedżera zadań*** Windows jest wyświetlana w kolumnie ***Status***ocena ryzyka (w skali od 1 do 9), czyli wskaźnik pokazujący prawdopodobieństwo tego, że wybrany proces to złośliwy kod.



Sposób prezentowania danych odzwierciedla hierarchiczną strukturę procesów. Zaznaczenie dowolnej nazwy w drzewie elementów po lewej stronie okna powoduje wyświetlenie wszelkich modułów powiązanych z wybranym procesem.



Informacje o procesach mogą być filtrowane nie tylko według szacowanego poziomu ryzyka, ale także według szczegółowości danych albo twórcy procesu. Standardowo program wyświetla dane o wszystkich procesach działających w systemie; można ten zestaw ograniczyć tylko do niepodpisanych przez Microsoft, wystarczy użyć skrótu klawiszowego [Ctrl]+[7].

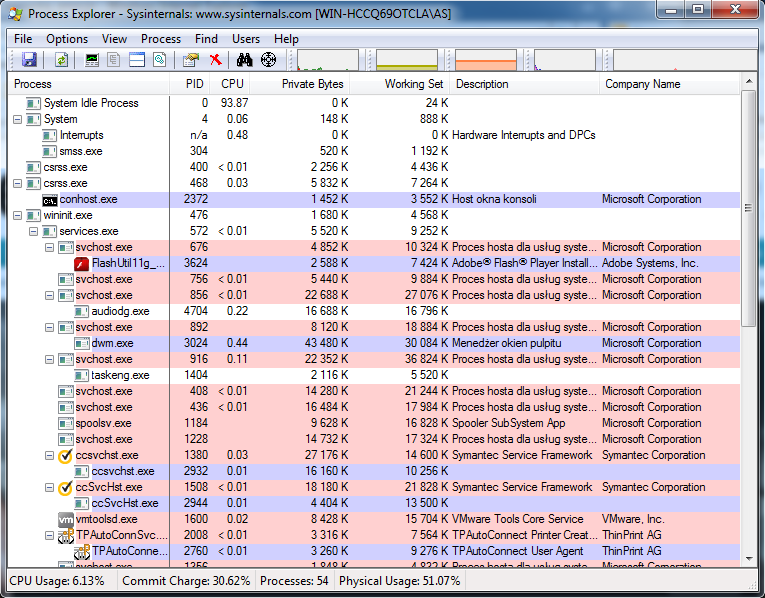


Okno dialogowe ***Log Statistic*** wyświetla dane ilościowe – posortowane według oceny ryzyka – dotyczące działających w  systemie elementów.

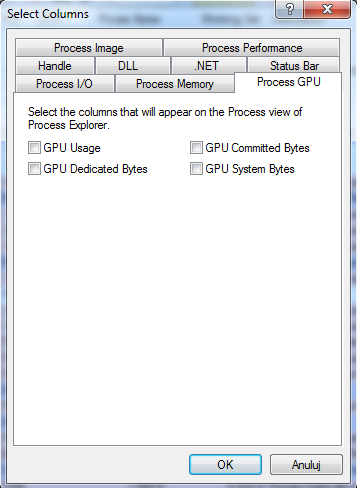
Dużą zaletą ***SysInspectora*** jest zdolność identyfikowania ukrytego kodu (rootkitów) w rejestrze systemu, sterownikach, procesach, usługach czy głównym rekordzie rozruchowym na dysku twardym. Program podczas każdego uruchomienia generuje logi i pozwala je później łatwo ze sobą porównywać.

[**Process Explorer**](http://pliki.onet.pl/Process.Explorer,Optymalizacja.systemu,Windows,11814.html)

***Process Explorer*** to kolejne narzędzie do zarządzania procesami w systemie Windows. Program autorstwa cenionego eksperta Marka Russinovicha jest zarazem przejrzystym i bardzo rozbudowanym narzędziem diagnostycznym, prezentującym szczegółowe informacje o tym, co działa w danej chwili w systemie i z jakimi elementami jest powiązane.

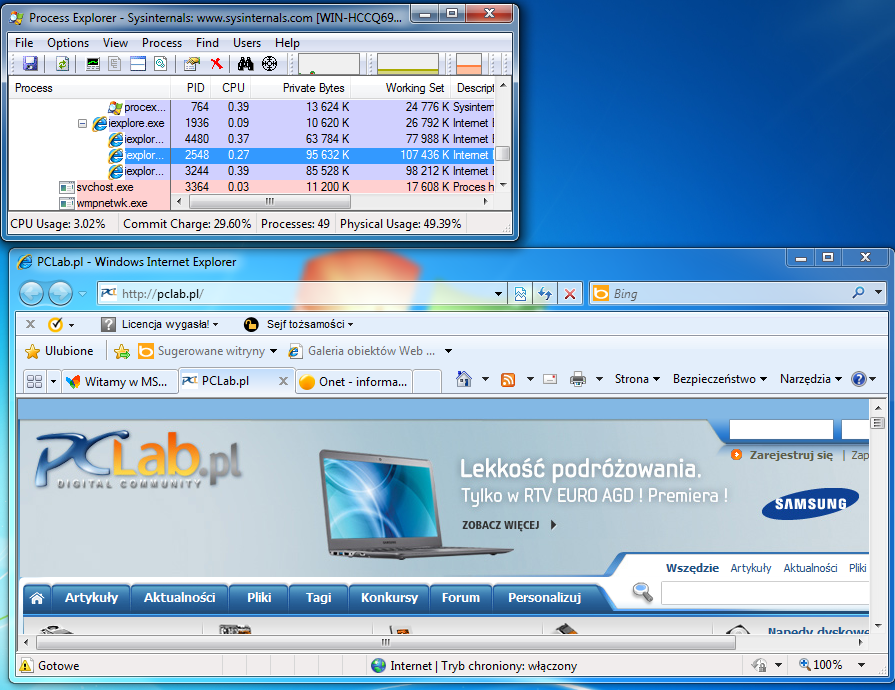


Program nie wymaga instalacji i dopasowuje się do wersji bitowej systemu operacyjnego. Po uruchomieniu ze standardowymi ustawieniami w oknie programu wyświetlane jest drzewo procesów i tabelaryczny zestaw podstawowych informacji, takich jak: nazwa procesu, identyfikator (PID), obciążenie procesora (CPU), dwa wskaźniki obciążenia pamięci (*PrivateBytes* oraz *Working Set*), opis (*Description*) oraz podpis cyfrowy firmy, która utworzyła dany program/proces (*Company Name*).



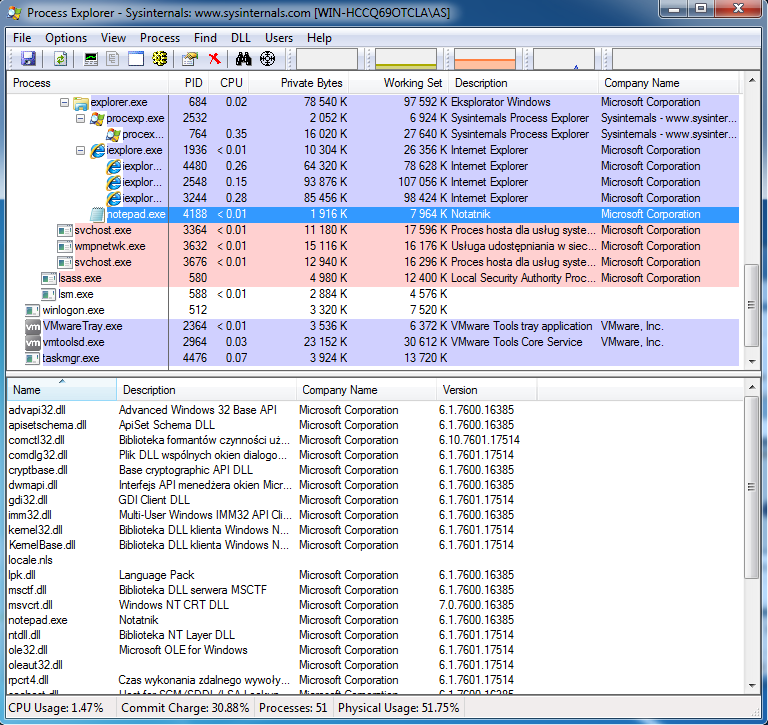
Standardowy zestaw danych można bardzo rozbudować, wybierając kolejne kolumny, których w przypadku ***Process Explorera*** jest znacznie więcej niż w ***Menedżerze zadań Windows***. Wybór danych do wyświetlenia jest ułatwiony dzięki temu, że dostępne pola wyboru reprezentujące poszczególne kolumny zostały podzielone na tematyczne kategorie, obejmujące m.in. informacje o pamięci (***Process Memory***), operacjach wejścia-wyjścia (***Process I/O***), a nawet operacjach wykonywanych z wykorzystaniem mocy obliczeniowej układów graficznych (***Process GPU***).

Przydatną funkcją programu jest możliwość łatwego ustalenia, jaki proces odpowiada za wybrane okno widoczne w systemie. Na przykład każda strona WWW widoczna na odrębnej karcie w przeglądarce Internet Explorer ma swój odrębny proces, który jest jednym z potomnych tego głównego – procesu samej przeglądarki. Na podstawie danych zgromadzonych w tabeli widocznej w oknie ***Process Explorera*** trudno rozpoznać, które procesy są związane z poszczególnymi kartami przeglądarki, ale bardzo ułatwi to narzędzie ***FindWindow'sProcess***.

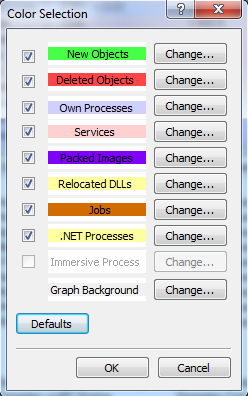


Wystarczy kliknąć ikonę z celownikiem widoczną na pasku ikon ***Process Explorera***, a następnie przeciągnąć ją na okno programu, którego proces ma zostać znaleziony. Jeśli celem jest zidentyfikowanie procesów poszczególnych kart w ***Internet Explorerze***, wystarczy nakierować wskaźnik na stronę WWW. (Uwaga: nie należy kierować go na samą kartę, bo to spowoduje, że ***Process Explorer*** zaznaczy proces główny przeglądarki, a nie proces danej karty!). Po wskazaniu strony w oknie ***ProcessExplorera*** zaznaczona zostanie odpowiednia pozycja na liście.

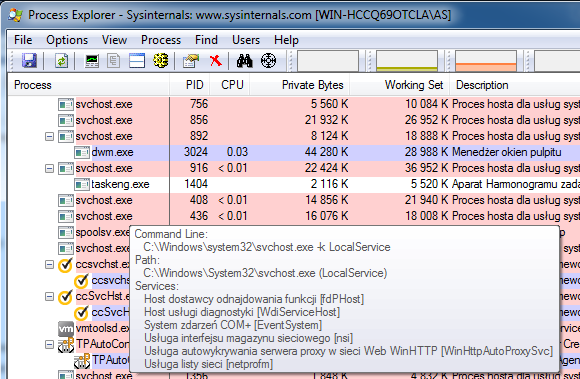
Za pomocą ***Process Explorera*** łatwo można sprawdzić, z jakich bibliotek DLL korzysta wybrany proces. Wystarczy kliknąć widoczną na pasku ikonę **ViewDLLs** lub użyć skrótu klawiszowego [Ctrl]+[D]. Okno programu zostanie wtedy podzielone na dwa panele i w dolnym zostanie wyświetlona lista bibliotek DLL powiązanych z zaznaczonym procesem z panelu górnego. Na poniższej ilustracji widoczny jest fragment listy bibliotek DLL powiązanych ze zwykłym ***Notatnikiem*** Windows.



Każdy wpis na liście procesów w oknie ***Process Explorera*** oznaczony jest odpowiednim kolorem. Łatwo się domyślić, że twórca aplikacji nie kierował się względami estetycznymi. Każdy kolor ma swoje znaczenie, które można odczytać w oknie dialogowym ***ColorSelection***. To pozwala łatwo odróżnić np. proces należący do zalogowanego użytkownika (***OwnProcesses***) od procesów usług systemowych (***Services***).

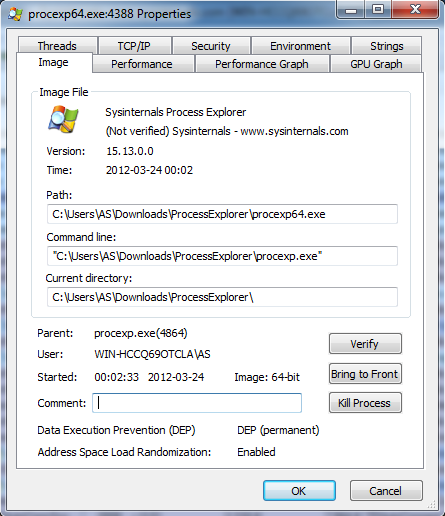


Dodatkowe informacje na temat poszczególnych procesów są udostępniane także w formie „dymku”, wyświetlanego po wskazaniu wybranego procesu.



Na powyższej ilustracji widoczne są informacje o jednej z licznych kopii ***svchost.exe***. Wyraźnie widać, z jakimi usługami systemowymi dany proces jest powiązany.

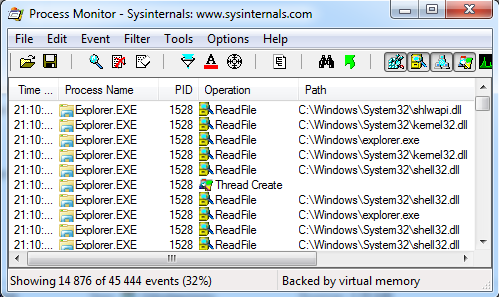
Jeszcze dokładniejsze dane o poszczególnych procesach można uzyskać, wywołując okno właściwości wybranego (należy w tym celu posłużyć się jego menu kontekstowym).



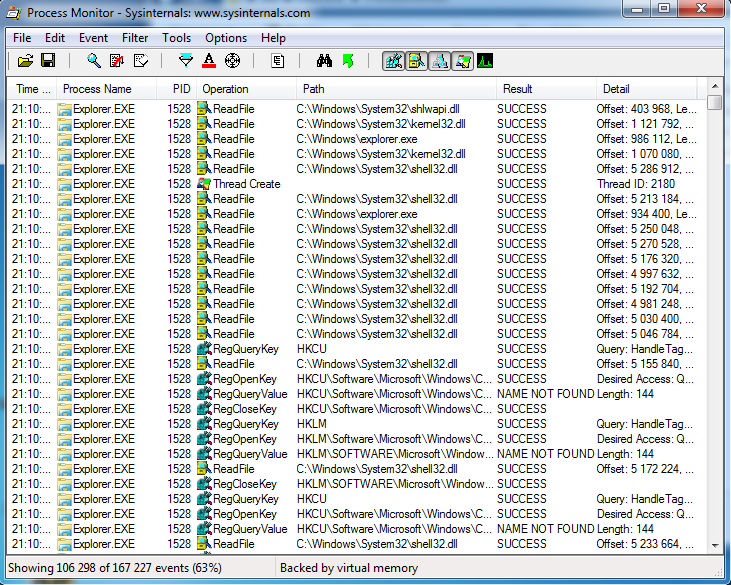
Każda z kart dostępnych w oknie właściwości procesu zawiera inny zestaw informacji. Standardowo aktywna jest karta ***Image***, zawierająca najważniejsze dane dotyczące wybranego obiektu. Można tu sprawdzić m.in. ścieżkę dostępu do pliku procesu (***Path***), polecenie, które utworzyło dany proces (***Commandline***), folder roboczy procesu (***Currentdirectory***), nazwę procesu nadrzędnego (***Parent***), nazwę konta użytkownika, który uruchomił dany proces (***User***), i czas działania procesu. Przyciskiem ***Bring to Front***można przywołać okno wybranego procesu na pierwszy plan. Jeżeli nie są z nim powiązane żadne okna, wyświetlony zostaje odpowiedni komunikat.

[**Process Monitor**](http://pliki.onet.pl/Process.Monitor,Informacje.i.diagnostyka,Windows,111296.html)

O ile ***Process Explorer*** prezentuje szczegółowe informacje o uruchomionych w systemie procesach, używanych przez nich zasobach oraz ich powiązaniach, to ***Process Monitor***, kolejne znakomite narzędzie, służy do monitorowania na bieżąco tego, co dzieje się w systemie.



Program działa na zasadzie rejestratora zdarzeń. Po uruchomieniu go okno aplikacji zaczyna wypełniać się przechwytywanymi informacjami. Liczba gromadzonych przez program danych jest imponująca. ***Process Monitor*** odnotuje niemal każdą zmianę w rejestrze systemowym czy systemie plików.



W standardowej konfiguracji ***Process Monitor*** wyświetla nazwę procesu inicjującego dane działanie, identyfikator procesu, rodzaj operacji, ścieżkę do obiektu, z którym powiązane jest zdarzenie (nie jest to ścieżka do procesu), rezultat operacji oraz dodatkowe informacje (w kolumnie ***Detail***).

Logi tworzone przez ***Process Monitor*** można zapisywać i dowolnie filtrować. Dzięki temu narzędzie bardzo ułatwia analizę zmian w systemie, np. wprowadzonych podczas instalacji jakiegoś programu w wersji próbnej. Taka wiedza pozwala później usunąć oprogramowanie bez pozostawienia jakichkolwiek śladów.

**Podsumowanie**

Wszystkie opisane metody zdobywania informacji o tym, co jest uruchomione w systemie, uzupełniają się wzajemnie. Wybór narzędzia to kwestia upodobań, wiedzy i sytuacji, z jaką przyjdzie się zmierzyć. Standardowy ***Menedżer zadań*** w zupełności wystarczy np. do sporadycznego przeglądania procesów czy ewentualnie zamykania wadliwie działających programów. Ale jego możliwości okazują się niewystarczające w sytuacji, gdy trzeba przejrzeć procesy na wielu komputerach albo zarządzać działaniem wielu z nich jednocześnie. W takich przypadkach znacznie lepiej użyć narzędzi działających w trybie tekstowym. Brak okienek i graficznego interfejsu jest rekompensowany możliwością zastosowania ich w skryptach. Szczególnie wart wyróżnienia jest tu ***Windows PowerShell***, który wymaga odpowiedniej wiedzy, ale też zapewnia ogromne możliwości.

Graficzne programy niezależne od systemu, takie jak: ***ESET SysInspector, Process Explorer, Process Monitor***, to rozwiązania pośrednie, optymalne dla tych, którym nie wystarczają możliwości zwykłego ***Menedżera zadań***, a którzy nie potrzebują narzędzi skryptowych.