

WINDOWS NT - THREAD SCHEDULER

Ha Nhat Viet

Adam Bułak

Damian Klata

23 grudnia 2005

Spis treści

Spis treści	2
1 Wprowadzenie	3
2 Pojęcia	3
2.1 Proces	3
2.2 Wątek	6
2.2.1 Stan	7
2.2.2 Priorytet	8
2.2.3 Kwant czasu	9
3 Struktury i funkcje	9
3.1 Baza Danych Dyspozytora(Dispatcher DataBase)	9
3.2 FindReadyThread	10
3.3 ReadyThread	10
3.4 ContextSwitch	10
4 Działanie planisty	11
4.1 Wygaśnięcie kwantu czasu wątku	11
4.2 Przejście wątku w stan oczekiwania	11
4.3 Przejście wątku w stan gotowości	11
4.4 Zmiana priorytetu wątku	11
5 Czynności specjalne	12
5.1 Wyjścia wątku z oczekiwania na operacje We/Wy	12
5.2 Po oczekiwaniu na semafor lub zdarzenie	12
5.3 Dla wątków pierwszoplanowych	12
5.4 Zapobieganie zagłodzeniu	13
6 System wieloprocessorowy	13
6.1 Obsługiwane systemy wieloprocessorowe	14
6.2 Baza danych dyspozytora - REAKTYWACJA	14
6.3 Wątek a procesor	14
6.4 Szukanie procesora	14
6.4.1 Są wolne procesory	14
6.4.2 Brak wolnych procesorów	15
6.5 Wybranie wątku do uruchomienia	15
7 Narzędzia systemowe	15
8 Bibliografia	16

1 Wprowadzenie

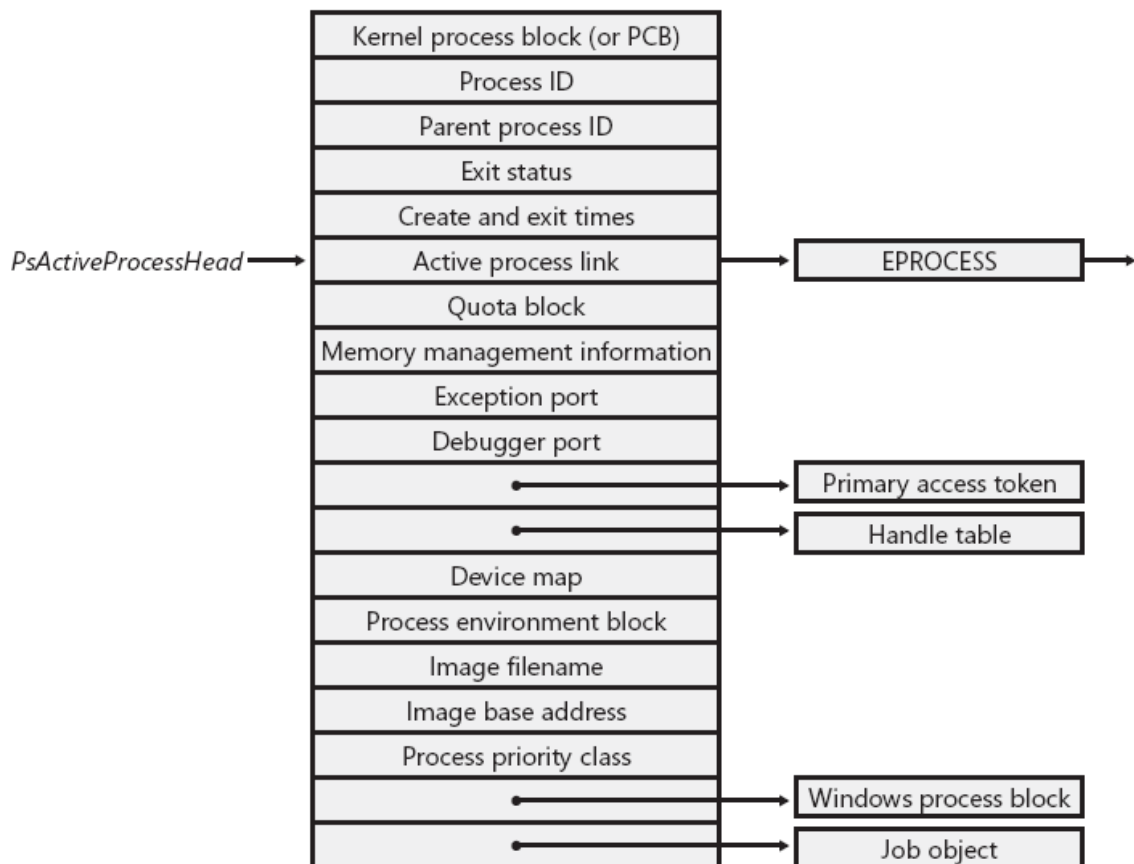
Windows NT jest wywłaszczającym, wielodostępowym systemem operacyjnym. Cecha wywłaszczania to inaczej możliwość odebrania procesora aktualnie wykonywującemu się zadaniowi. W taki sposób system zapewnia wszystkim uruchomionym zadaniom dostęp do zasobów procesora. Wszystkie systemy operacyjne oparte na Windows NT(New Technology) takie jak obecne Windows 2000, Windows XP, Windows Server 2003 oraz systemy będące jeszcze w fazie konstrukcji jak Windows XP X64, Windows Vista i Windows Longhorn udostępniają również możliwość podziału i szeregowania zadań na komputerach wieloprocessorowych.

2 Pojęcia

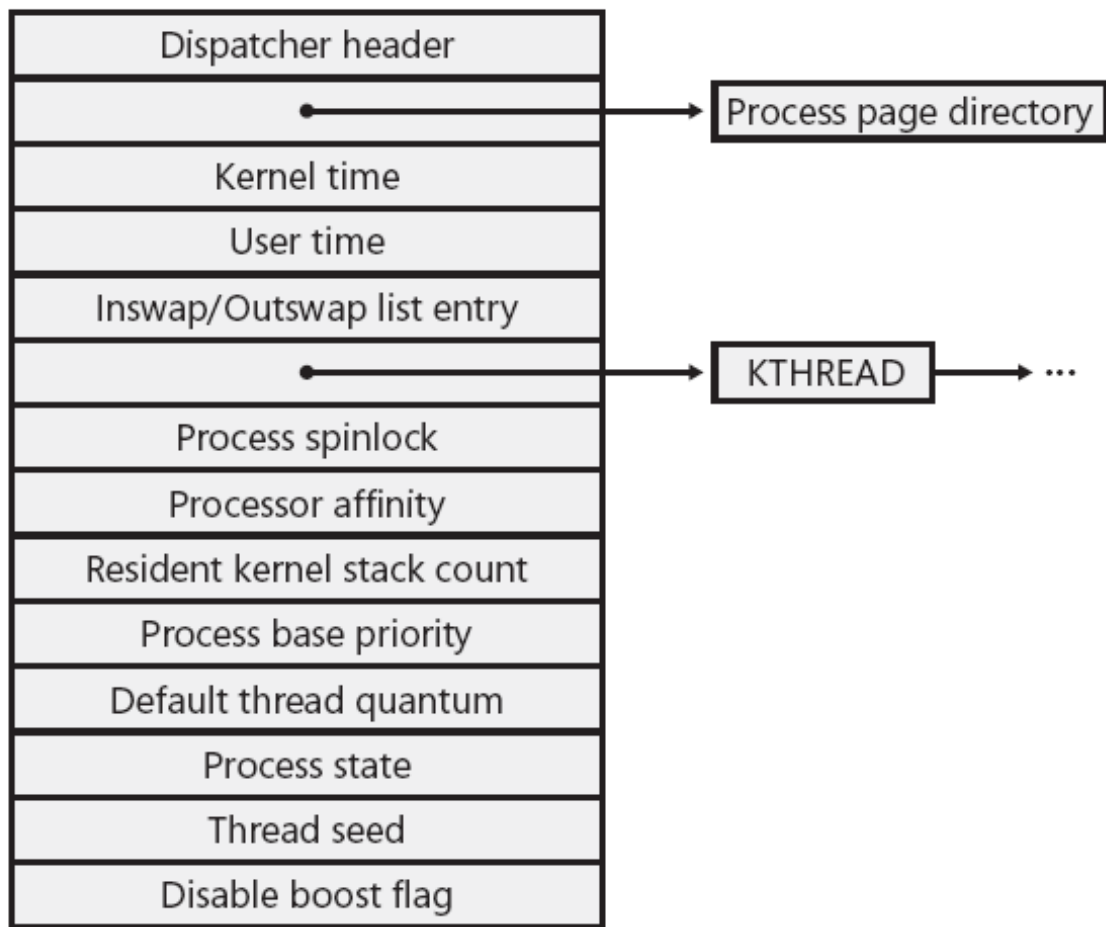
Poniżej zostaną przedstawione dwa fundamentalne pojęcia, niezbędne do zrozumienia działania całego systemu.

2.1 Proces

Proces jest to egzemplarz uruchomionego programu. Zadaniem i celem procesu jest wykonywać kod programu w danym systemie. Informacje na temat procesów są zapisane w jądrze systemu. Każdy proces jest reprezentowany przez blok uruchomionego procesu (EPROCESS). Blok procesu zawiera wszystkie informacje na temat procesu i oprócz tego określa również zależności między procesem a innymi strukturami, między innymi - wątkami. Poniżej znajduje się obraz opisujący strukturę EPROCESS.



Do przestrzeni pamięci jądra należą prawie wszystkie pola tej struktury. Przechowywany w przestrzeni użytkownika jest wyłącznie blok środowiska procesu. Planista korzysta jednak głównie z treści zawartego w bloku procesu jądra. Blok ten zwany inaczej blokiem kontrolnym procesu zawiera podstawowe informacje na temat sposobu przydzielania zadań.



- Dispatcher header - wskaźnik do kolejki oczekujących, o którym wspomnę jeszcze nie raz
- KTHREAD list - lista wszystkich wątków należących do tego procesu
- Processor affinity - wykorzystywane w komputerach wieloprocessorowych do określenia na których procesorach może być proces wykonywany
- Process base priority - bazowy priorytet, z którego będą korzystały wątki
- Default thread quantum - domyślna długość kwantu czasu

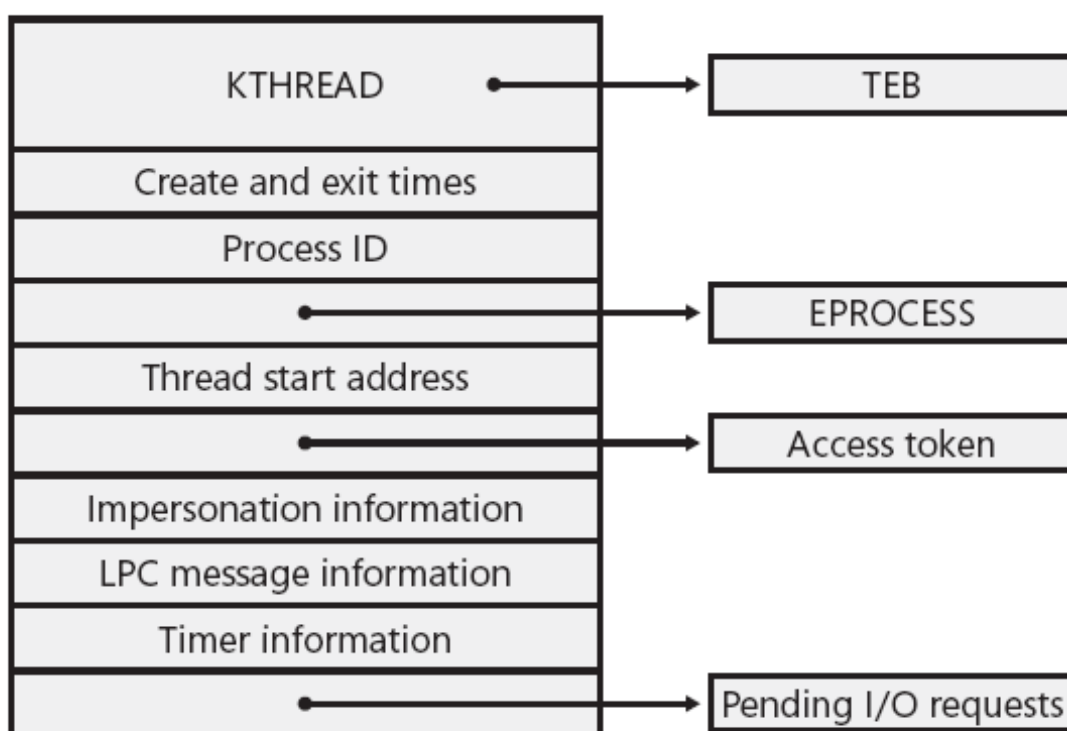
Proces może zostać stworzony za pomocą funkcji `CreateProcess`, która przyporządkowuje osobie wywołującej funkcję stworzony proces oraz jego wątek. Przy tworzeniu nowego procesu wszystkie wartości są dziedziczone po procesie macierzystym.

W skład dokumentu wchodzi opis poszczególnych przypadków użycia, który dla każdego z nich będzie zawierał cel, problemy związane z wydajnością przeprowadzania tego przypadku

użycia, warunki wstępne niezbędne do rozpoczęcia operacji, opis czynności składających się na jej przeprowadzenie (podstawowe oraz alternatywne), a także dodatkowe informacje (takie jak przyporządkowanie przypadku użycia, ryzyko z nim związane czy możliwe usprawnienia).

2.2 Wątek

Po umieszczeniu już procesu w przestrzeni adresów, wywoływana jest funkcja `NTCreateThread`, która tworzy wątek zgodny z obrazem procesu. Wątek jest częścią procesu mającą własne zadanie do wykonania. Każdy wątek należy dokładnie do jednego procesu. Struktura wątku podobnie jak i procesu trzymana jest w głównej przestrzeni jądra. Również i wątek posiada swój blok jądra zwany `KThread block`, w którym przechowuje informacje na temat swojej pracy.



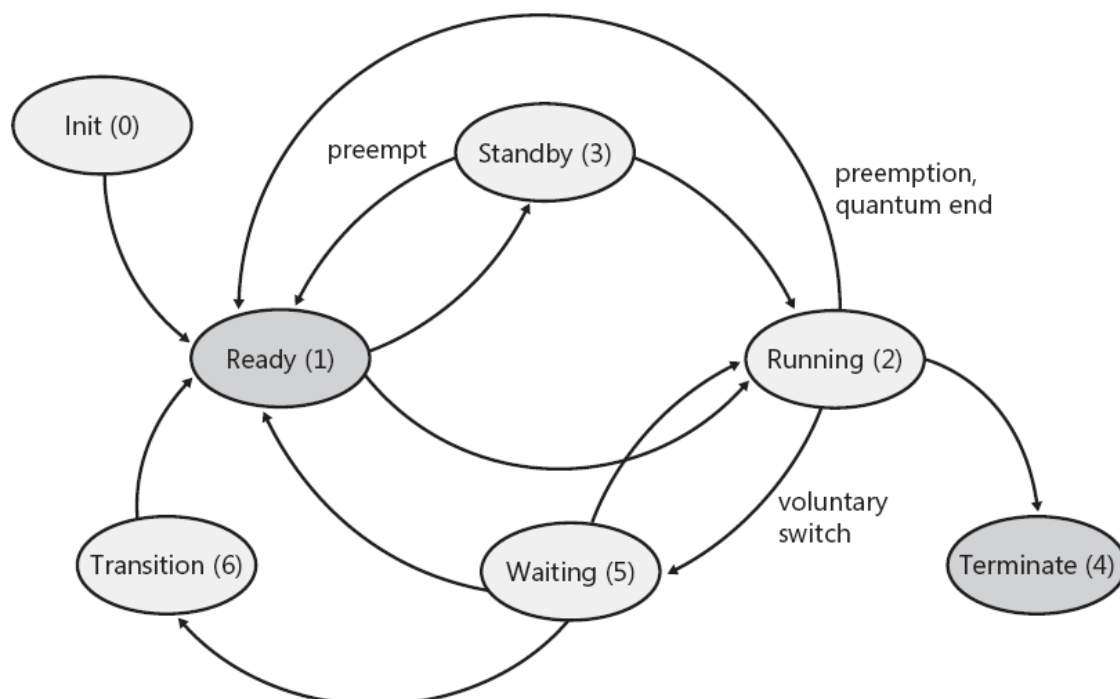
Wątek po stworzeniu jest w stanie inicjalizacji. W tym stanie wywoływana jest funkcja `KeInitThread`, która ustawia wszystkie zmienne w strukturze wątku. Wątek większość wartości takie jak: priorytet bazowy, długość kwantu czasu, maskę procesorów dziedziczy po procesie. Po zainicjowaniu wszystkiego wątek zmienia stan na gotowy i od tego momentu bierze udział w rywalizacji o procesor. Poniżej znajdują się najważniejsze pojęcia dotyczące wątku, istotne dla dalszych objaśnień.

- Stan
- Priorytet
- Kwant czasu

- Maska bitowa procesorów

2.2.1 Stan

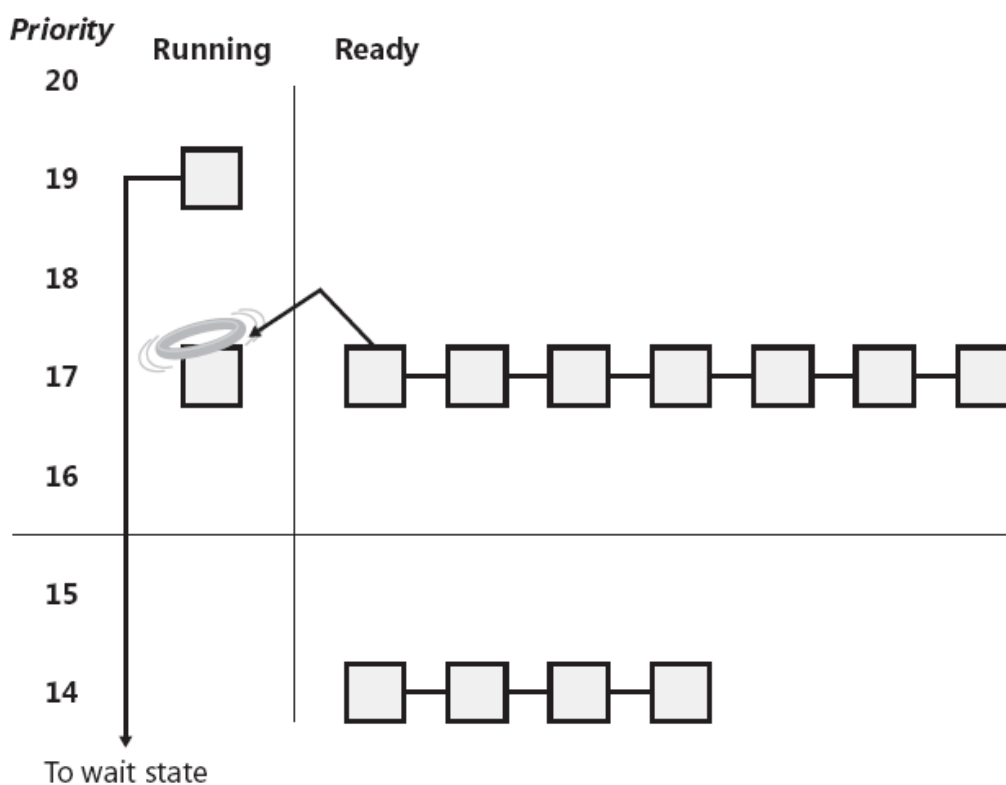
Stan jest jednym z własności opisujących wątek. W windows 2000 i Windows XP do opisanie wątku służy 7 stanów. W Windowsie Server 2003 zbiór ten został wzbogacony o jeden dodatkowy stan.



- Inicjalizowany(Initialized) - Świeżo po utworzeniu, atrybuty wątku nie są jeszcze ustalone
- Gotowy(Ready) - Wątek czeka uruchomienie.
- Następny(Standby) - Wątek został wybrany jako następny do uruchomienia na danym procesorze. Każdy procesor może wybrać wyłącznie 1 taki wątek.
- Działający(Running) - Wątek posiada procesor
- Oczekujący(Waiting) - Wątek oczekuje na jakieś zdarzenie
- Przejściowy(Transition) - Wątek jest gotowy do uruchomienia, kontekst wątku nie został załadowany do pamięci jądra
- Zlikwidowany(Terminated) - Wątek wykonał swoje zadanie i oczekuje na zniszczenie.
- Odroczone(Deferred ready) - W Windows Server 2003 na komputerze wieloprocessorowym. Wątek został przypisany procesorowi ale nie został zakolejkowany.

2.2.2 Priorytet

W Windowsie NT mamy 32 priorytetów(0 to najniższy a 31 najwyższy) Priorytety dzielą się też na 2 poziomy. Poziom dynamiczny obejmuje priorytety od 1 do 15 a poziom czasu rzeczywistego od 16 do 31. Priorytet zerowy nie może być przydzielony żadnemu procesowi i może go mieć tylko proces bezczynności, który pętlę się poszukując wątku do uruchomienia. Każdy wątek zapisuje w swojej strukturze 2 priorytety: priorytet bazowy i priorytet chwilowy. Priorytet bazowy jest dziedziczony z procesu macierzystego i jest niezmienny. Pierwotna wartość priorytetu chwilowego jest ustawiana na wartość priorytetu bazowego. Na zmianę wartości priorytetu pierwotnego wpływa planista. Zmienić priorytet bazowy może tylko proces uprzywilejowany. Głównym czynnikiem przy podejmowaniu decyzji szeregowania wątków jest priorytet.



W Windows NT jest również podział na klasy priorytetów - czasu rzeczywistego, wysoki, ponad normalny, normalny, poniżej normalnego i bezczynny. Istnieje również stopniowanie priorytetów względem priorytetu bazowego(krytyczny,najwyższy,ponad normalny,normalny,poniżej normalnego,najniższy,bezczywny). Modyfikatory najwyższy(+2), ponad normalny(+1), normalny(+0), poniżej normalnego(-1), najniższy(-2) zmieniają priorytet wątku o odpowiednią wartość. Modyfikatory krytyczny(15 i 31) i bezczynny(1 i 16) ustalają wartość priorytetu na wartości graniczne dla danego poziomu priorytetów.

2.2.3 Kwant czasu

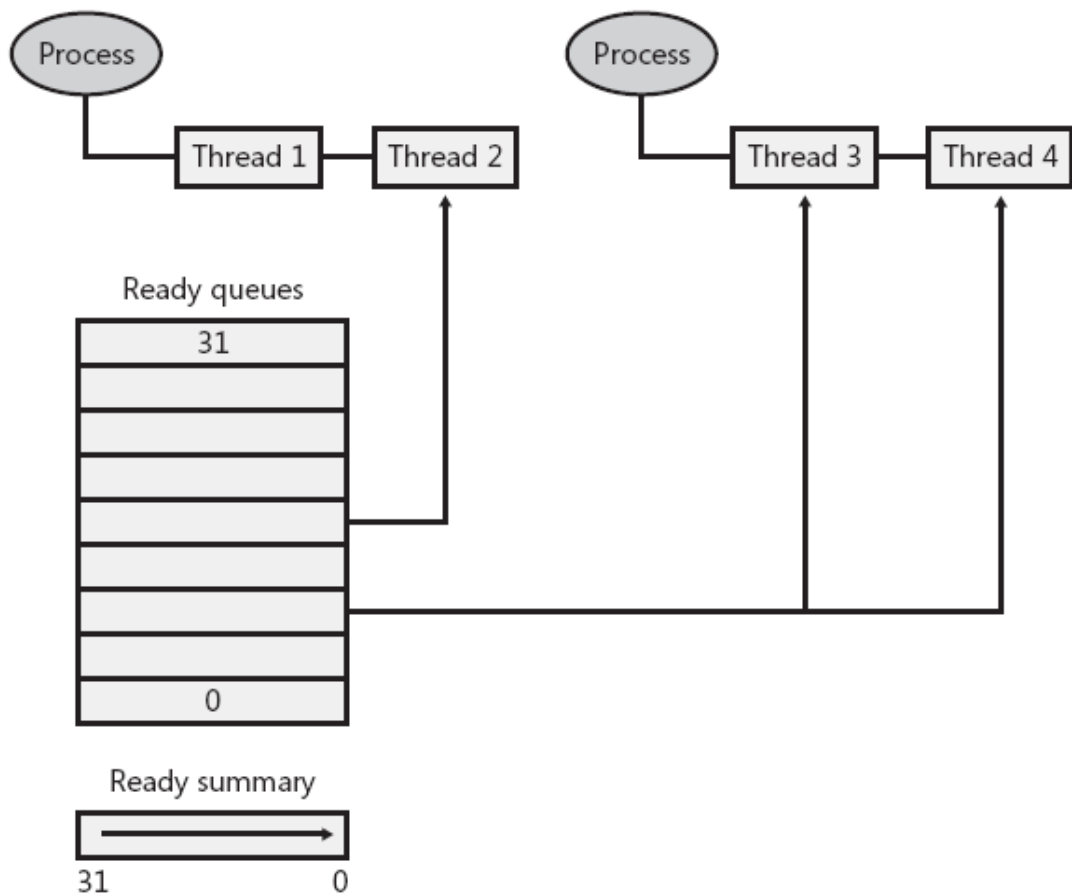
Kwant czasu jest to okres czasu jaki wątek dostaje na wydzierżawienie procesora. Kwant czasu jest liczony w wielokrotnościach interwału. Interwał jest to pewna jednostka czasu charakterystyczna dla architektury komputera. Dla komputerów jednoprocessorowych z serii x86 wartość ta wynosi najczęściej 10ms. Na komputerach wieloprocessorowych ta wartość wynosi 15ms. Obecnie często się jednak spotyka architektury jednoprocessorowe z interwałem równym 15ms. Kwant czasu jest wartością najczęściej z zakresu od 2(komputery osobiste) do 12 interwałów(serwery). Można również własnoręcznie zmienić domyślną długość kwantu czasu. Jednakże przy ręcznej modyfikacji dostępne są tylko opcje 2 lub 12 interwałów. W strukturze wątków kwant czasu nie jest jednak zapisany po prostu jako wielokrotność interwałów lecz jako potrójna wartość tej wielokrotności. Powodem takiego zapisu jest sposób w jaki system Windows NT redukuje długość kwantu czasu. Długość kwantu czasu zapisanego w strukturze jest zredukowana zawsze po upłygnięciu jednego interwału. Wątek może jednak zasnąć w oczekiwaniu na zdarzenie przed upłygnięciem jednego interwału i przez to nie zostanie mu zmniejszona wartość. Aby temu zapobiec wprowadzono dodatkową redukcję wartości kwantu czasu po wyjściu ze stanu oczekiwania. Przy każdym takim wyjściu wartość licznika kwantu czasu jest zmniejszany o 1 czyli teoretycznie o 1/3 długości interwału(nazywane dalej jednostkami kwantowymi). Do dokumentu dołączony będzie słownik terminów używanych w opisach.

3 Struktury i funkcje

3.1 Baza Danych Dyspozytora(Dispatcher DataBase)

Baza danych dyspozytora przetrzymuje informacje na temat oczekujących na procesor wątków, oraz na temat które wątki są wykonywane przez jakie procesory. W systemach Windows 2000 i Windows XP baza danych dyspozytora w systemie jednoprocessorowym nie różni się od bazy systemu wieloprocessorowego. W obu przypadkach istniała jedna lista gotowych wątków, w której umieszczane były wszystkie oczekujące wątki(niezależne od procesora). W systemie Windows Server 2003 każdy procesor posiada własną listę.

- Lista gotowych wątków - 32 elementowa tablica kolejek gotowych wątków. W komórce i-tej znajduje się kolejka(zaimplementowana w postaci listy) gotowych wątków o priorytecie i.
- Mapa bitowa gotowych kolejek - Maska bitowa mówiąca które kolejki wątków nie są puste.



3.2 FindReadyThread

Funkcja ta zwraca wątek gotowy do wykonania. W systemie uniprocessorowym jest to pierwszy wątek z niepustej kolejki o najwyższym priorytecie.

3.3 ReadyThread

Ta Funkcja zgłasza gotowość do wykonania. W systemie uniprocessorowym wątek który wywołał tą funkcję zostaje porównywany do obecnie wykonującego się wątku i jeśli oczekujący wątek ma wyższy priorytet to następuje zjawisko wywłaszczenia. Wywłaszczenie polega na oddaniu procesora wątkowi oczekującemu, zaś wątek który został wywłaszczony trafia na początek kolejki oczekujących. Jeśli wątek nie ma wystarczająco dużego priorytetu to musi jednak ustawić się w kolejce gotowych.

3.4 ContextSwitch

Powyższa funkcja przełącza kontekst wątku w jądrze. Następuje to poprzez zachowanie kontekstu (wskaźnik wykonywanej instrukcji, wskaźniki do stosów, wskaźnik do przestrzeni adresowej)

aktualnie wykonującego się wątku w stosie wątków w jądrze. W strukturze wątku zapisywany jest wskaźnik do stosu, a do pamięci jądra ładowany jest kontekst nowego wątku. Jeśli dany wątek należy do innego procesu niż wcześniej wykonywany, do pamięci jądra ładowana jest przestrzeń adresowa procesu.

4 Działanie planisty

Zadaniem planisty jest uszeregować dostępy wszystkich wątków do procesora, uwzględniając odpowiednio priorytety, stany i poszczególne sytuacje wątków. Jego podstawowym obowiązkiem jest więc zareagować na następujące zdarzenia i orzec który wątek powinien być wybrany na następnika do wykonania:

4.1 Wygaśnięcie kwantu czasu wątku

Kiedy wątek wyczerpie swoje ostatnie tyknięcie zegara(interwał), zabierany jest od niego procesor w poszukiwaniu następnika. Wątek po wyjściu z procesora trafia na koniec kolejki o swoim priorytecie. Procesor zaś wywołuje funkcję FindReadyThread. Może oczywiście zajść sytuacja, iż ten sam wątek dostanie zasoby procesora(następuje to w przypadku gdy jest to wątek o najwyższym priorytecie).



4.2 Przejście wątku w stan oczekiwania

Sytuacja podobna do powyższego przypadku. Wątek zawiesza się w oczekiwaniu na zdarzenie. Za pomocą funkcji FindReadyThread znajdujemy nowy wątek. Następuje przełączenie kontekstu.

4.3 Przejście wątku w stan gotowości

Otrzymawszy zasób/zdarzenie na który oczekiwał wątek, wątek wraca do stanu gotowości by znowu móc korzystać z zasobów procesora. Taki wątek po zakończeniu okresu oczekiwania wywołuje funkcję ReadyThread, przy czym musi zmniejszyć sobie licznik kwantu czasu o jedną jednostkę(nie robią tego wątki o priorytecie większym niż 13).

4.4 Zmiana priorytetu wątku

Istnieją sytuacje(co zostanie opisane później), w których wątek zmienia swój priorytet. System Windows NT uwzględnia tę możliwość i przy każdej zmianie priorytetu wątku gotowego

do wykonywania sprawdza czy nowy priorytet oczekującego wątku nie przewyższa priorytetu wykonującego się wątku. Jeśli tak to następuje wywłaszczenie procesora znane z funkcji Ready-Thread.

5 Czynności specjalne

W celu zachowania równowagi w podziale procesora pomiędzy wątkami system Windows NT stworzył możliwość przyznawania specjalnych przywilejów, dotyczących się wartości priorytetu oraz kwantu czasu wątku. Przywileje te przyznawane są w sytuacjach:

5.1 Wyjścia wątku z oczekiwania na operacje We/Wy

Windows NT podobnie jak inne systemy operacyjne ma skłonność do faworyzowania procesów interaktywnych, gdyż te rzadziej korzystają z procesora. To ukazuje się w specjalnym traktowaniu wątków korzystających z operacji na We/Wy. Każdy wątek po zakończeniu czekania na We/Wy dostaje automatyczny przyrost priorytetu. Przyrost ten zastosowany jest jednak względem wartości bazowej a nie obecnej(chwilowej). Przyrost ten jest też tylko czasowy i zmniejsza się o 1 co kwant czasu aż osiągnie wartość bazową. Wartość priorytetu może się zwiększyć o dowolną ilość ale nie może wyjść poza poziom(1-15,16-31) priorytetu do którego należy priorytet bazowy.

5.2 Po oczekiwaniu na semafor lub zdarzenie

Podobnie jak w przypadku oczekiwania na operacje We/Wy i tutaj wątek otrzymuje specjalny przyrost priorytetu. Istnieje jednak w tym przypadku jeszcze jeden specjalny bonus dla wątków które zostaną obudzone przez zdarzenia z specjalną funkcją NtSetEventBoostPriority. Przy takich okazjach priorytet wątku jest ustawiony na wartość zapisaną w funkcji NtSetEventBoostPriority zwiększoną o 1. A kwant czasu wykonywania jest zmieniona na $\max(4, \text{obecna_wartosc})$ jednostek kwantowych.

5.3 Dla wątków pierwszoplanowych

Wątek jest pierwszoplanowy gdy został wybrany do interakcji z użytkownikiem. Taki wątek jest preferowany przez system i też dostaje specjalne bonusy w postaci zwiększenia priorytetu i kwantu czasu:

W momencie kiedy wątek pierwszoplanowy kończy etap oczekiwania, jego priorytet chwilowy jest zwiększany o wartość zapisaną w PsPrioritySeparation. Opcji ta jest systemowa i nie da się jej w żaden sposób wyłączyć.

W systemach NT przed 4.0 przywilej każdego wątku pierwszoplanowego polegał na zwiększeniu wartości priorytetu(o 2). Ta metoda często jednak generowała zagłodzenie procesów w tle i przez to był nieskuteczny. W następnych systemach NT rozwiązano ten problem poprzez zwiększenie wartości kwantu czasu. Do opisanego sposobu modyfikacji wartości kwantu czasu dla

wątku pierwszoplanowego służy rejestr HKLM \SYSTEM \CurrentControlSet \Control \PriorityControl \Win32PrioritySeparation , który ma następującą strukturę(6-bitową):

		Short			Long		
Variable	6	12	18	12	24	36	
Fixed	18	18	18	36	36	36	

- Długość - To pole określa czy kwant czasu ma być krótki(6 interwałów) czy długi(12 interwałów). Z 4 możliwych wartości 1 oznacza długi, 2 krótki a 0 i 3 wartość domyślna dla systemu(patrz Kwant Czasu).
- Czy stałe - Czy ma istnieć możliwość wpłynięcia na mnożnik przedłużenia czasu wykonywania dla wątków pierwszoplanowych. 1 znaczy tak, 2 znaczy nie, przy czym dla stała wartość mnożnika wynosi 3. Jeśli w tym polu jest wartość 0 lub 3 to znaczy, że ma być domyślnie jak w systemie(dla Windows XP i 2000 możemy ustalić mnożnik, w 2003 Server wartość jest stała).
- Mnożnik - Jak długi ma być kwant czasu dla wątku pierwszoplanowego. Wartości od 0-2 przedstawiają wartości odpowiednie do wyżej przedstawionej ilustracji. Wartość 3 jest błędna.

5.4 Zapobieganie zagłodzeniu

Podstawowy scenariusz działania planisty nie zapobiega zagłodzeniom. Tym zadaniem zajmuje się specjalny mechanizm zwany "balance set manager". "Balance set manager" jest to system, który budzi się co sekundę i sprawdza stan wszystkich wątków. Kiedy system ten zauważy, że jakiś wątek przez długi okres(trzy lub cztery sekundy, w zależności od systemu) przebywał w stanie gotowości i nie dostał procesora to wówczas może stwierdzić, że dany wątek został zagłodzony i powinien dostać ódszkodowanie". System rewanżuje się takiemu wątkowi poprzez zwiększenie jego priorytetu do 15 oraz wydłużenie jego kwantu czasu dwukrotnie(w Windowsach XP i 2000) lub do 4 jednostek kwantowych(w Windows Server 2003). Zmiana nie jest jednak aplikowana na stałe. Po wyczerpaniu się kwantu czasu wszystkie atrybuty zmienione przez balance set manager'a zostają przywrócone.

6 System wieloprocesorowy

Powyższy opis dotyczy działania planisty w komputerach uniprocessorowych. Rzecz ma się inaczej w systemie wieloprocesorowym, gdzie wątki mogą być przypisane różnym procesorom. Problem jest tym większy gdy platforma jest ASMP(Asymmetric Multiprocessor). Wówczas system przy rozdziale musi jeszcze wziąć pod uwagę preferencje poszczególnych wątków. Całość ta zostanie opisana poniżej, przy użyciu terminów oraz informacji zawartych w poprzednich punktach.

6.1 Obsługiwane systemy wieloprocesorowe

Oprócz zwykłego systemu wieloprocesorowego Windows NT obsługuje jeszcze dwa bardzo popularne systemy multiprocessingowe:

- NUMA(Non-Uniform Memory Access) - System wieloprocesorowy, gdzie oprócz wspólnego obszaru pamięci są jeszcze obszary pamięci dla poszczególnych procesorów. Procesory tego systemu są grupowane do tzw. wierzchołków(ang. node) z własną pamięcią. Dostęp do pamięci wewnątrz wierzchołka jest o wiele szybszy niż do pamięci wspólnej.
- Hyperthreading - System wieloprocesorowy, który potrafi na jednym procesorze fizycznym symulować wiele procesorów logicznych. Szeregowanie zadań dla tych systemów istotnie różni się od szeregowania na normalnym systemie wieloprocesorowym, co zostanie przedstawione później.

6.2 Baza danych dyspozytora - REAKTYWACJA

Baza danych dyspozytora w systemach wieloprocesorowych nie różni się wiele od tej dla jednoprocessorowej. Różnica występuje(jak to zostało opisane wcześniej) jedynie w Windows Server 2003. Dla systemów Windows XP i 2000, które mają identyczną strukturę bazy danych dla architektury jednoprocessorowej i wieloprocesorowej powstał jednak nowy problem, związany z dostępem do pamięci. Dlatego na komputerach wieloprocesorowych dostęp do bazy danych dyspozytora chronią wirujące blokady().

6.3 Wątek a procesor

Windows NT umożliwia wątkowi wybór procesorów, które mają go obsługiwać. Do tego służy zapisana w procesie maska bitowa upodobań(ang. affinity mask). W tej masce bitowej są zapisane na jakich procesorach mogą się wątki danego procesu wykonywać. Systemy Windows NT gwarantują, że wątek nie zostanie uruchomiony na procesorze, którego nie lubi. Maska bitowa upodobań może zostać zdefiniowana zarówno dla procesu - wszystkich jego wątków, czy też dla pojedynczego wątku.

Oprócz maski upodobań wątek posiada również w swojej strukturze dwa pola : ideal i last, określające odpowiednio ulubiony i ostatni procesor.

6.4 Szukanie procesora

Kiedy wątek staje się gotowy do uruchomienia, znajduwany jest mu procesor na którym mógłby zostać uruchomiony.

6.4.1 Są wolne procesory

Jeśli są jakieś wolne(zgodne z maską upodobań) procesory to na pewno wątek zostanie uruchomiony na którymś z tych procesorów. Wątek szereguje procesory w kolejności : ulubiony, ostani

uruchomiony, obecnie używany (ten na którym wykonuje się kod planisty). W przypadku gdy żaden z powyższych procesorów nie jest wolny to:

1. w systemie 2000 wybierany jest procesor o największym numerze
2. w XP i 2003, przy NUMA zmniejszamy nasz zbiór do przecięcia z wierzchołkiem (jeśli ma wolne procesory), w którym znajduje się nasz ulubiony procesor
3. w hyperthreading redukcja polega na wybraniu procesora fizycznego w którym wszystkie procesory logiczne są wolne. Jeśli nie ma takiego procesora, to szukany jest zbiór w którym jest jakiś bezczynny procesor (szukający procesora dla wątku) zawierający ulubiony lub ostatnio wykonywany procesor.

6.4.2 Brak wolnych procesorów

Jeśli nie ma wolnego procesora to następuje próba uruchomienia wątku na jego ulubionym procesorze. Scenariusz dopasowania wątku do procesora wygląda następująco: Wątek próbuje wywłaszczyć uruchomiony wątek. Jeśli jego priorytet na to nie pozwala to próbuje on wywłaszczyć następny wątek do uruchomienia (każdy procesor posiada 1 taki wątek). W przypadku gdy i to jest niemożliwe to wątek spoczywa w kolejce gotowych.

6.5 Wybranie wątku do uruchomienia

Windows XP i 2000

W przeciwieństwie do trybu uniprocessorowego tutaj system nie wybiera po prostu wątku o najwyższym priorytecie z kolejki, lecz wątek o najwyższym priorytecie spełniającym jedno z 4 następujących warunków:

- Jego ulubionym procesorem jest dany procesor
- Dany procesor jest ostatnim procesorem na którym się wykonywał
- Czekał dłużej niż 3 długości kwantu czasu
- Ma priorytet większy niż 24

W Windows 2003 wieloprocessorowy tryb wyboru wątku jest identyczny do wyboru w trybie jednoprocessorowym.

7 Narzędzia systemowe

Windows NT udostępnia pewne struktury i narzędzia do monitorowania pracy procesów i wątków. Najbardziej znanym i najprostszym narzędziem jest z pewnością Menedżer Zadań. Za pomocą tego programu można (tylko z przywilejami superużytkownika) zmienić priorytet bazowy uruchomionego procesu. Dla osób chcących mieć większą kontrolę nad wątkami i procesami

uruchomionymi w systemie, twórcy Windows NT stworzyli liczniki". Dla każdego procesu i wątku system przechowuje szereg liczników monitorujących przebieg ich działań. Liczniki zapisują między innymi moment uruchomienia procesu, czas dzierżawienia procesora, stan wątku, priorytety wątków i procesów. Dostęp do tych liczników można otrzymać poprzez wywołanie programu "perfmon".

8 Bibliografia

- Microsoft Windows Internals
- www.sysinternals.com
- Wikipedia na en.wikipedia.org
- MS Windows NT Thread Scheduling na <http://www.808multimedia.com/winnt/index.htm>
- Inside the Windows NT Scheduler