

Windows Scheduler

Teraz przedstawimy schemat szeregowania procesów w systemie operacyjnym Windows®



Windows Scheduler

Nie ma jednego modułu czy algorytmu odpowiedzialnego za szeregowanie procesów(a dokładnie ich wątków).

Szeregowanie jest zaimplementowane w jądrze i nazywa się ogólnie „dispatcher”.



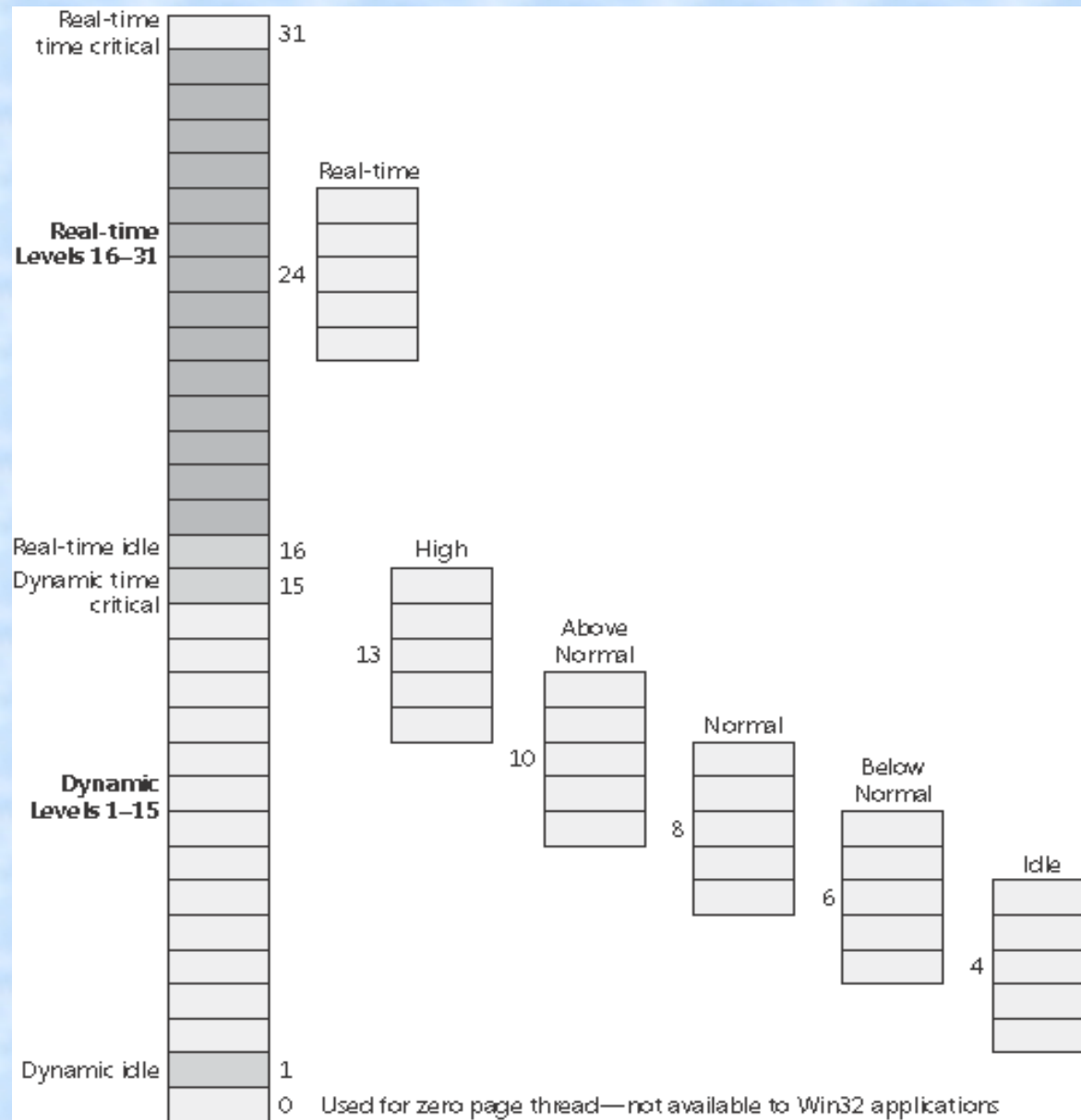
Windows Scheduler

Następujące zdarzenia mogą wywołać „dispatching”:

- Wątek jest gotowy do uruchomienia
- Wątek kończy status uruchomienia, ponieważ kończy kwant czasu, kończy, sam oddaje procesor, przechodzi w stan czekania.
- Priorytet wątku zmienia się.



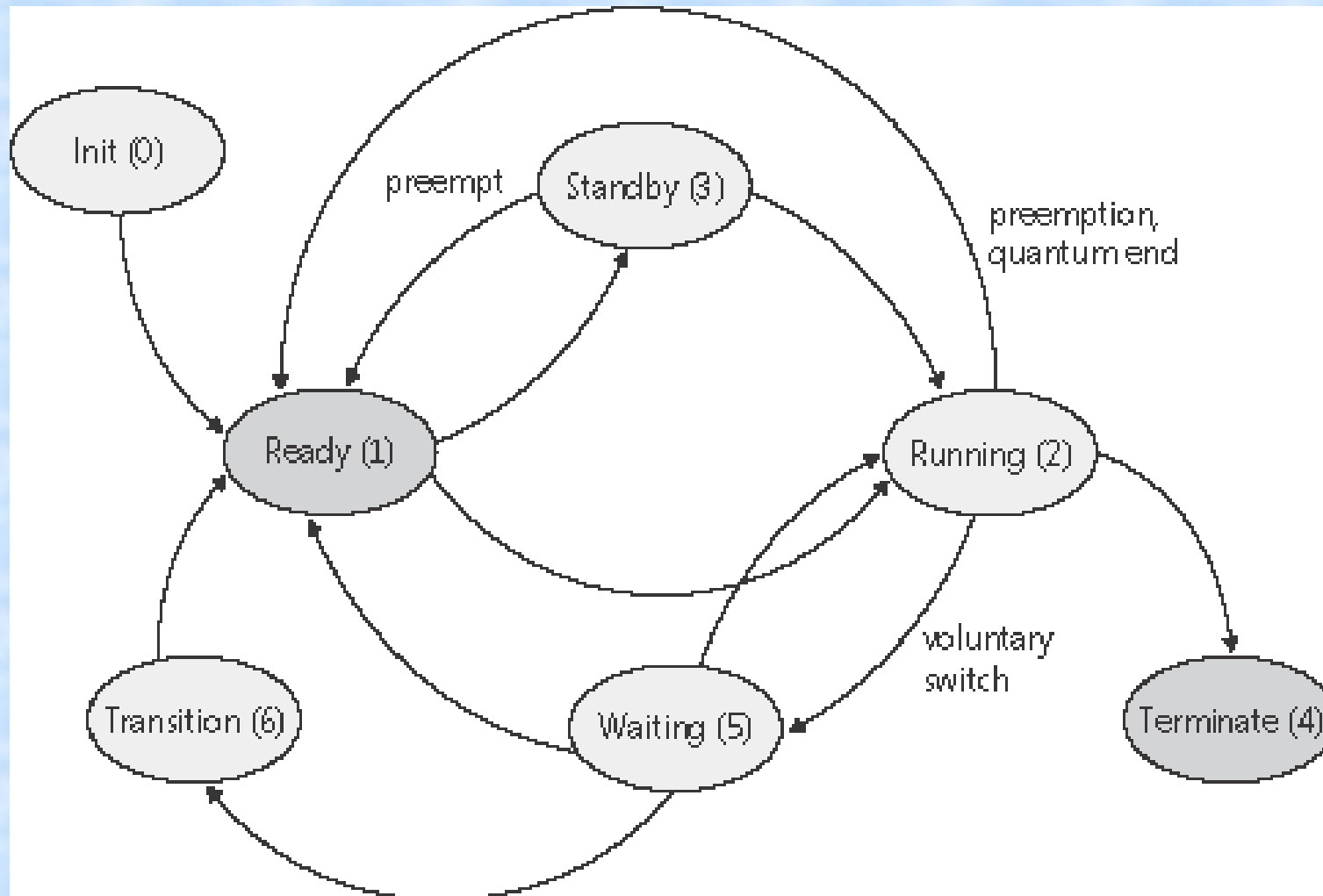
Priorytety



Jak widać na przedstawionym obrazku system Windows definiuje sześć klasy priorytetów: Czasu Rzeczywistego, Wysoki, Powyżej normalnego, Normalny, Poniżej normalnego oraz Niski.



Stan wątku procesu w Windows



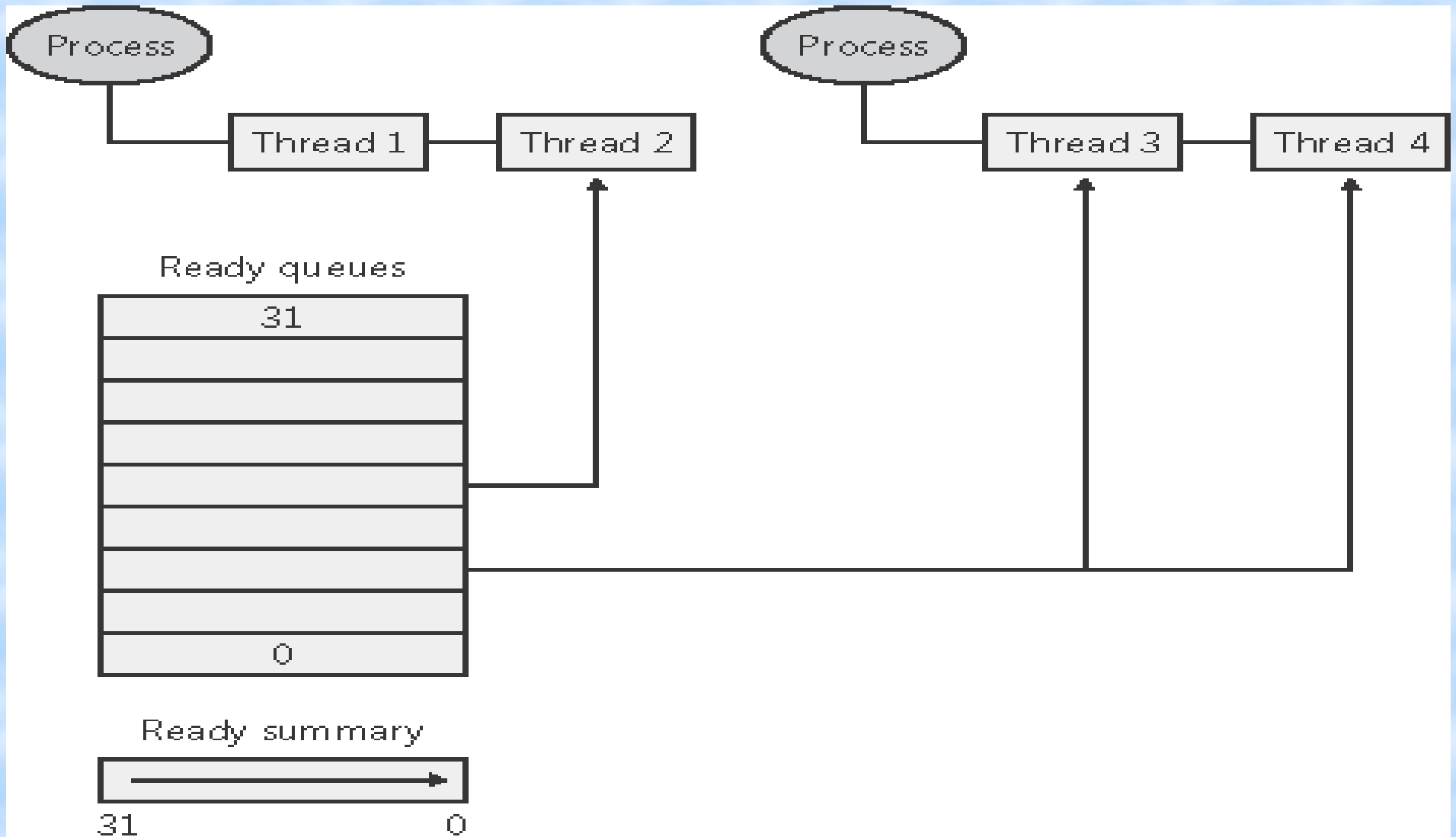
Szukanie gotowych wątków

Do znalezienia gotowych wątków wykorzystywane są dwie zmienne w jądrze systemu:

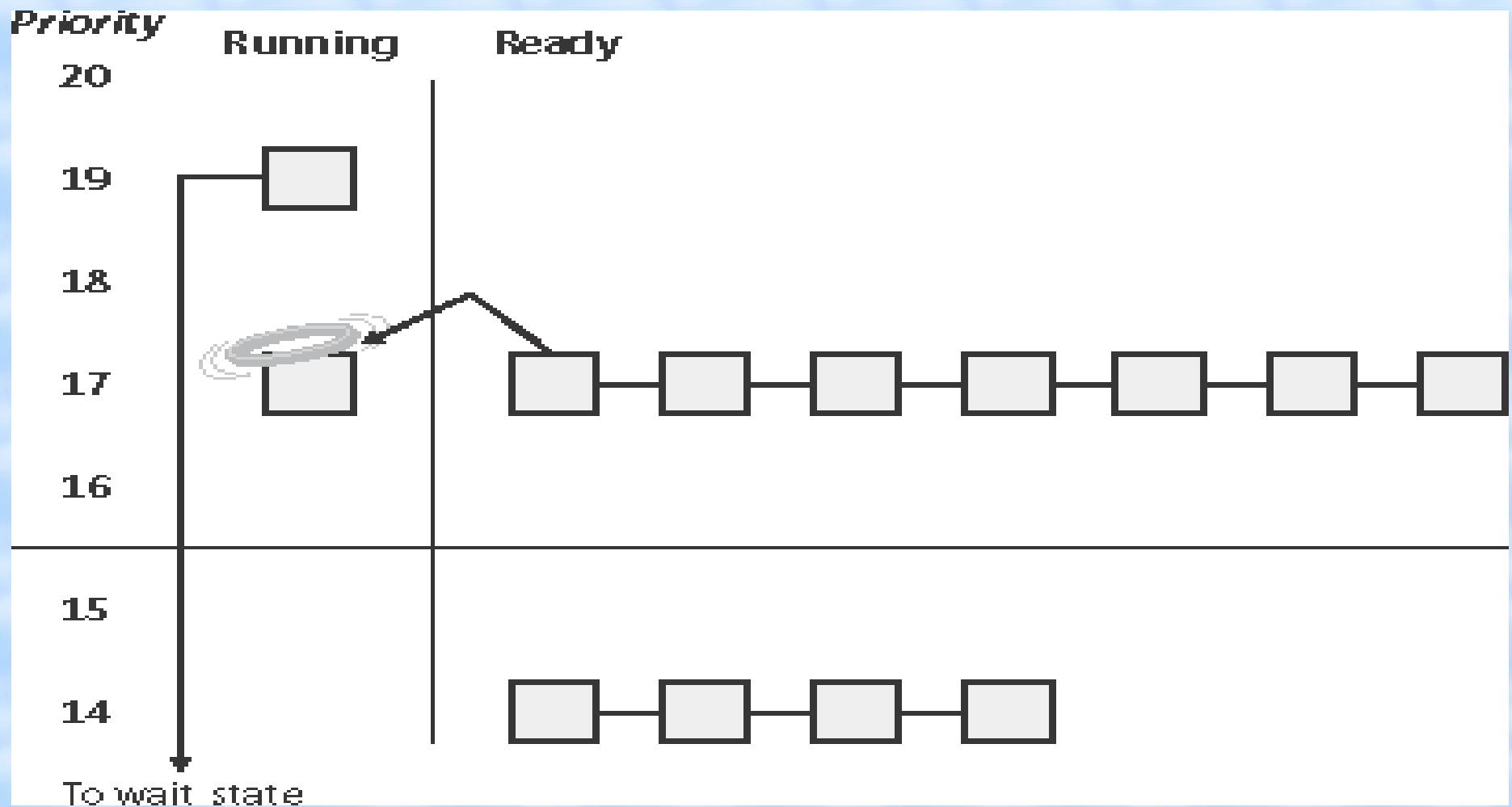
- KiReadySummary jest to bitmapa dla priorytetów które mają chociaż jeden gotowy do uruchomienia wątek.
- KiDispatcherReadyListHead jest to tablica 32 kolejek. Każda kolejka zawiera wątki gotowe do użycia dla pojedynczego priorytetu.



Dispatcher Database



Oddanie procesora

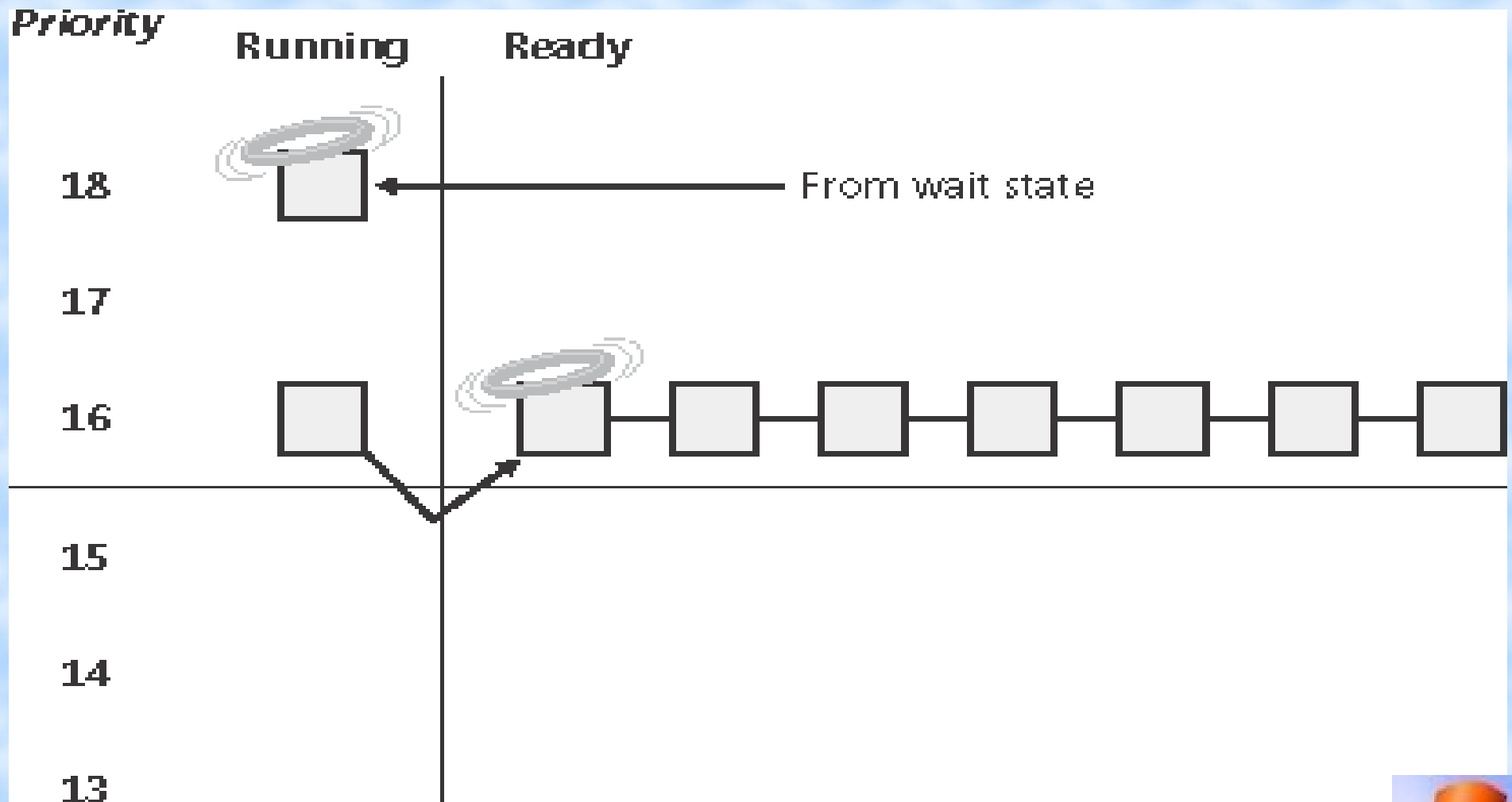


Wywłaszczenie

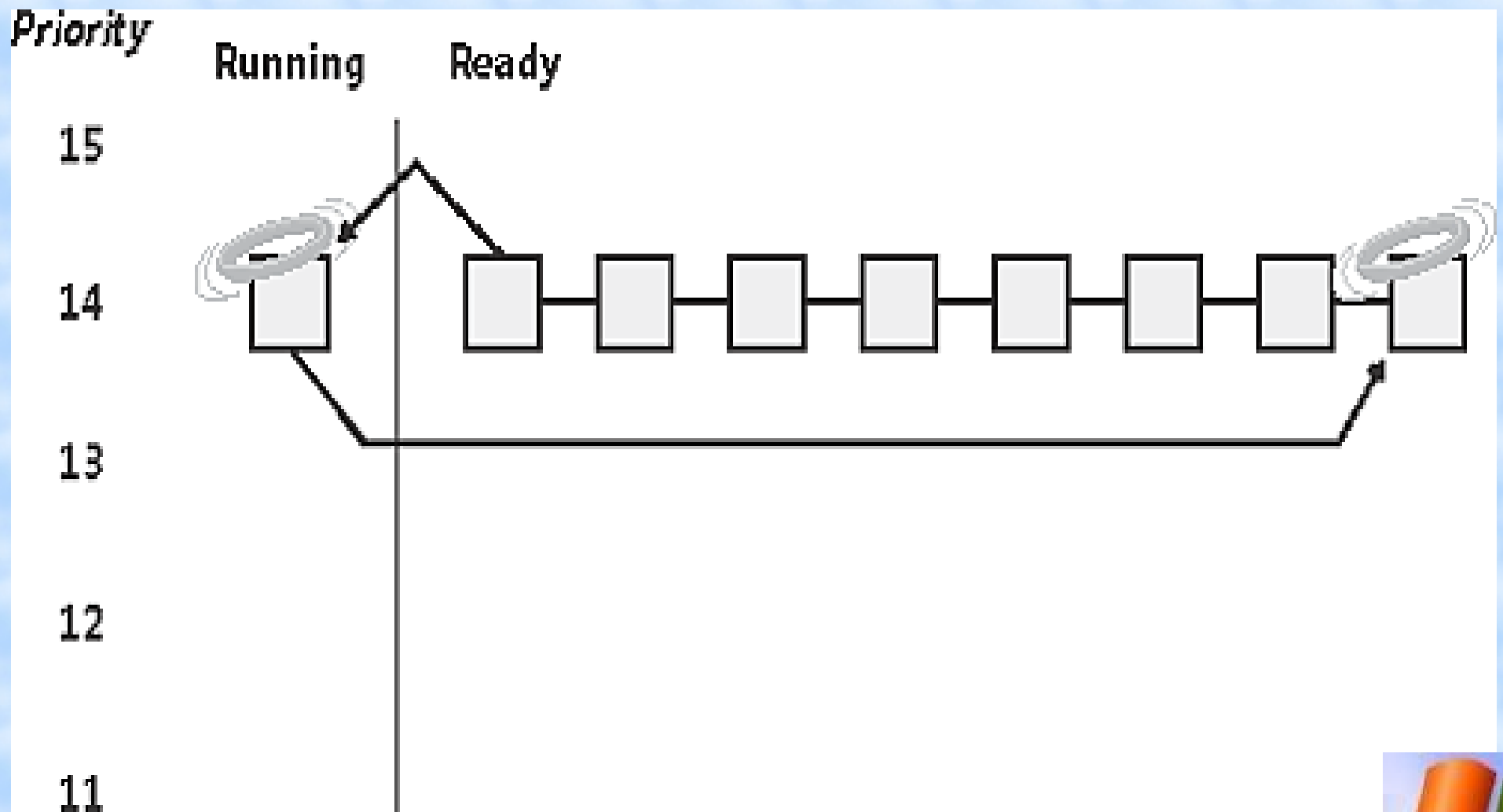
Wątek może zostać wywłaszczony w dwóch przypadkach:

- Wątek o wyższym priorytecie zakończy czekanie.
- Zmieni się priorytet wątku

Wywłaszczanie procesu



Koniec kwantu czasu



Kwant czasu

Kwant czasu przyznawany jest jako wielokrotność „clock intervals”(tyknięć zegara).

- W przypadku Windowsa XP są to dwa tyknięcia zegara, a dla np. Windows Server dwanaście tyknięć.



Quantum Value

Każdy proces posiada pole quantum value, które jest wykorzystywane do odliczania jego kwantu czasu.



Kontrola kwantu czasu

4	2	0
Short vs. Long	Variable vs. Fixed	Foreground Quantum Boost

Pole rejestru

HKLM\SYSTEM\CurrentControlSet\Control\PriorityControl\Win32PrioritySeparation definiuje właściwości kwantu czasu.



Boosting

Priorytety procesów mogą ulec zmianie poprzez przyspieszanie (ang. Boosting). Dotyczy to tylko zwykłych procesów, gdyż priorytety procesów czasu rzeczywistego mogą być zmieniane tylko przez program, nie przez system. Jednak maksymalną wartością priorytetów nadal jest 15.



Boosting

Istnieje pięć momentów gdy Windows może zwiększyć priorytet procesu:

- Zakończenie operacji wejścia – wyjścia
- Koniec oczekiwania np. na semaforze
- Jeśli interaktywny proces skończy czekać
- Jeśli wątki GUI zostaną aktywowane
- Zagłódzenie procesu



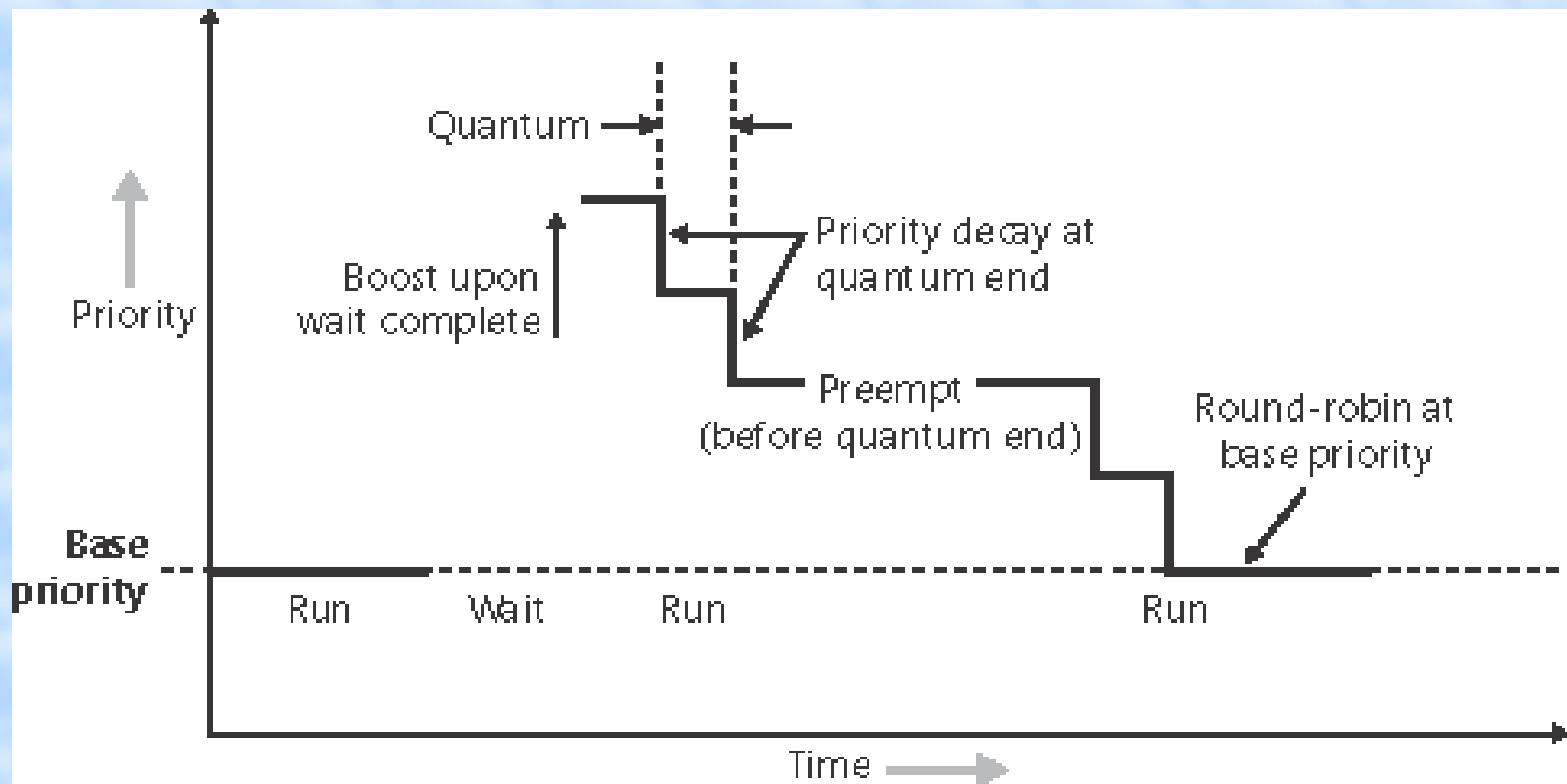
Boosting

Przykłady akceleracji sprzętowej

- 1 – Dysk, CD-ROM
- 2 – Sieć
- 6 – Mysz, klawiatura
- 8 - Dźwięk



Boosting



Zagłódzenie procesów

Raz na sekundę włączany jest Balance Set Manager, który przeszukuje kolejki gotowych wątków i szuka nie odpalanych przez około cztery sekundy.



Multiprocessorowość

Do obsługi wątków nadal wykorzystujemy dispatcher database. Jednak posiadamy dwie nowe zmienne

- Activeprocessormask – mapa bitowa procesorów
- Idlesummary – mapa bitowa wolnych procesorów



Jednoczesny dostęp

Aby uniknąć jednoczesnego dostępu
wykorzystujemy dwa spinlocki:

- KiDispatcherLock
- KiContextSwapLock



„Pokrewieństwo”

Każdy wątek posiada maskę procesów na których może być odpalany.



Idealny procesor

Każdy wątek posiada pole Ideal processor i last processor



Algorytmy szeregowania

- Wybieranie procesora dla gotowego wątku
- Wybieranie wątku dla procesora



Wybieranie procesora dla gotowego wątku

- Są „puste procesory”
- Nie ma pustych procesorów



Wybieranie wątku dla procesora

Poszukiwanie wątku spełniającego jeden z poniższych warunków

- Był ostatnio odpalony na tym procesorze
- Jest to jego idealny procesor
- Czeka na wykonanie od trzech tyknień zegara
- Jego priorytet jest większy od 24



Linux vs Windows

- Open source, dowolne zmiany kodu
- Niedoskonały RT
- Proces, który korzysta głównie z procesora, a raz na jakiś czas czeka na I/O, bonusu nie dostaje.
- Niezmienny kod
- Niedoskonały RT
- Jeśli proces czeka I/O dostaje bonus



Linux vs Windows

- Aby wprowadzić zmiany parametrów, trzeba rekompilować jądro
- Oddzielne kolejki dla każdego procesora.
- Część parametrów można zmienić poprzez aplikację lub rejestr. Reszty nie można.
- Globalna struktura dla wielu procesorów.

