

Systemy wejścia-wyjścia

Systemy wejścia-wyjścia (I/O)

- Sprzęt wejścia-wyjścia (I/O)
- Użytkowy interfejs wejścia-wyjścia (I/O)
- Podsystem I/O w jądrze S.O.
- Zamiana żądań operacji I/O na operacje sprzętowe
- Wydajność

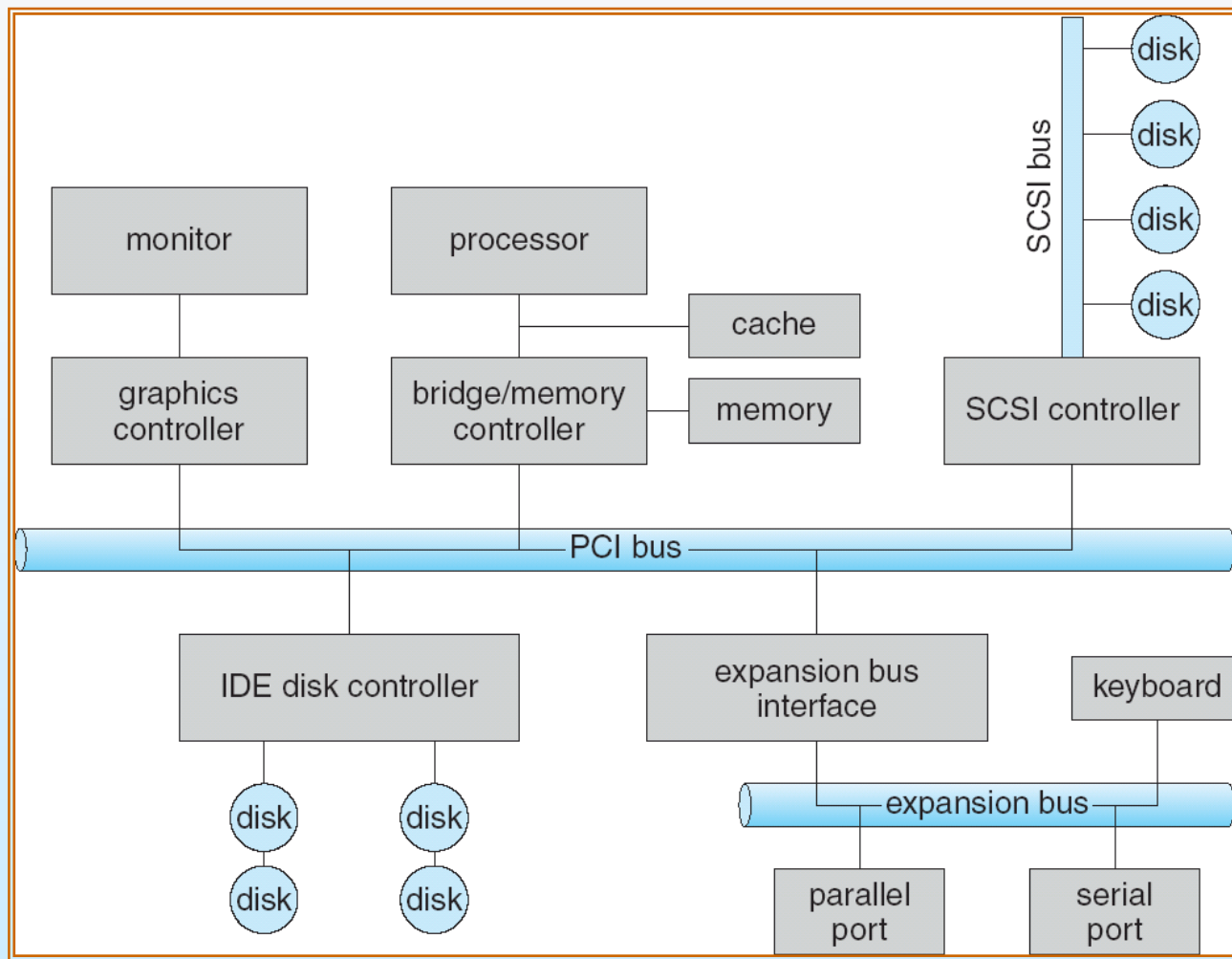
Tematyka wykładu

- Prezentacja struktury podsystemu wejścia wyjścia (I/O)
- Omówienie zasad sprzętu I/O i jego złożoności
- Przedstawienie aspektów wydajności sprzętu i oprogramowania wejścia-wyjścia

Sprzęt I/O

- Duża różnorodność urządzeń I/O
- Podstawowe pojęcia
 - **Port**
 - **Magistrala (daisy chain** lub dzielony dostęp bezpośredni)
 - **Sterownik (host adapter)**
- I/O instructions control devices
- Urządzenia posiadają adresy używane przez
 - Instrukcje bezpośredniego dostępu I/O
 - Wejście- wyjcie **Odwzorowane do pamięci (Memory-mapped)**

Typowa struktura struktura z magistralą



Wybrany zestaw portów urządzeń w PC

zakres adresów I/O (heksadecymalnie)	urządzenie
000–00F	kontroler DMA
020–021	kontroler przerwań
040–043	timer
200–20F	kontroler gier
2F8–2FF	port szeregowy (dodatkowy)
320–32F	kontroler dysku twardego
378–37F	port równoległy
3D0–3DF	kontroler grafiki
3F0–3F7	kontroler stacji dyskieta
3F8–3FF	port szeregowy (podstawowy)

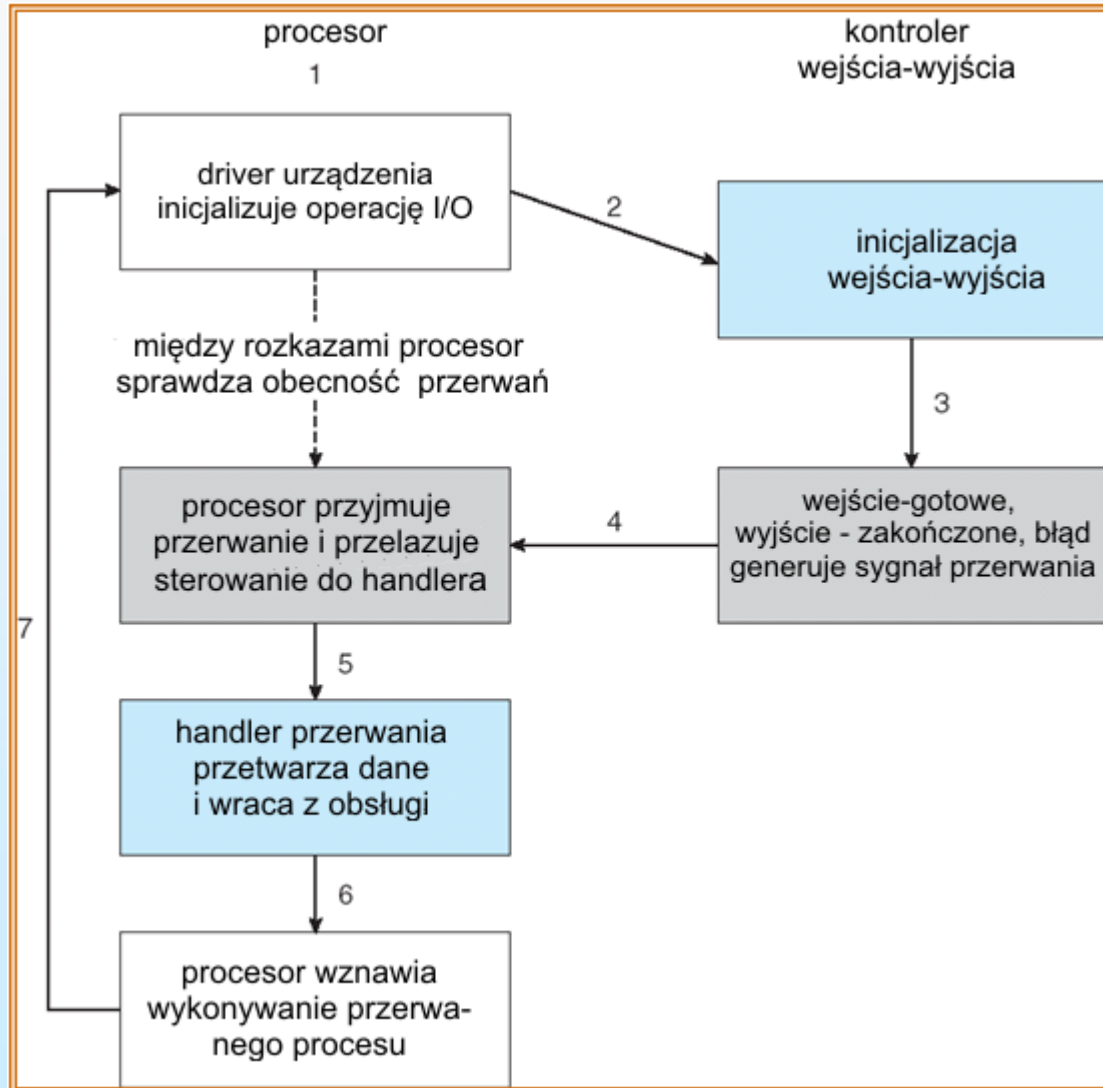
Odpytywanie –aktywne czekanie

- Określa stan urządzenia
 - gotowy (command-ready)
 - zajęty
 - Error
- Wykorzystanie cyklu „**Busy-wait**” do oczekiwania na zakończenie pracy przez urządzenia we/wy

Przerwania

- **Linia żądania przerwanie Procesora (CPU Interrupt-request line)** aktywowana przez urządzenie I/O
- **Program obsługi przerwania (Interrupt handler)** przejmuje przerwanie
- **Przerwanie maskowalne** to przerwanie, którego zgłoszenie można zablokować programowo (może służyć do ignorowani lub opóźniania przerw)
- **Wektor Przerwań (Interrupt vector)** przekazuje przerwanie właściwemu programowi obsługi przerwania
 - Bazujący na priorytetach przerwania
 - Niektóre przerwania są **niemaskowalne**
- Mechanizm przerw wykorzystywany jest także dla obsługi wyjątków (exceptions)

Przerwanie -cykl I/O



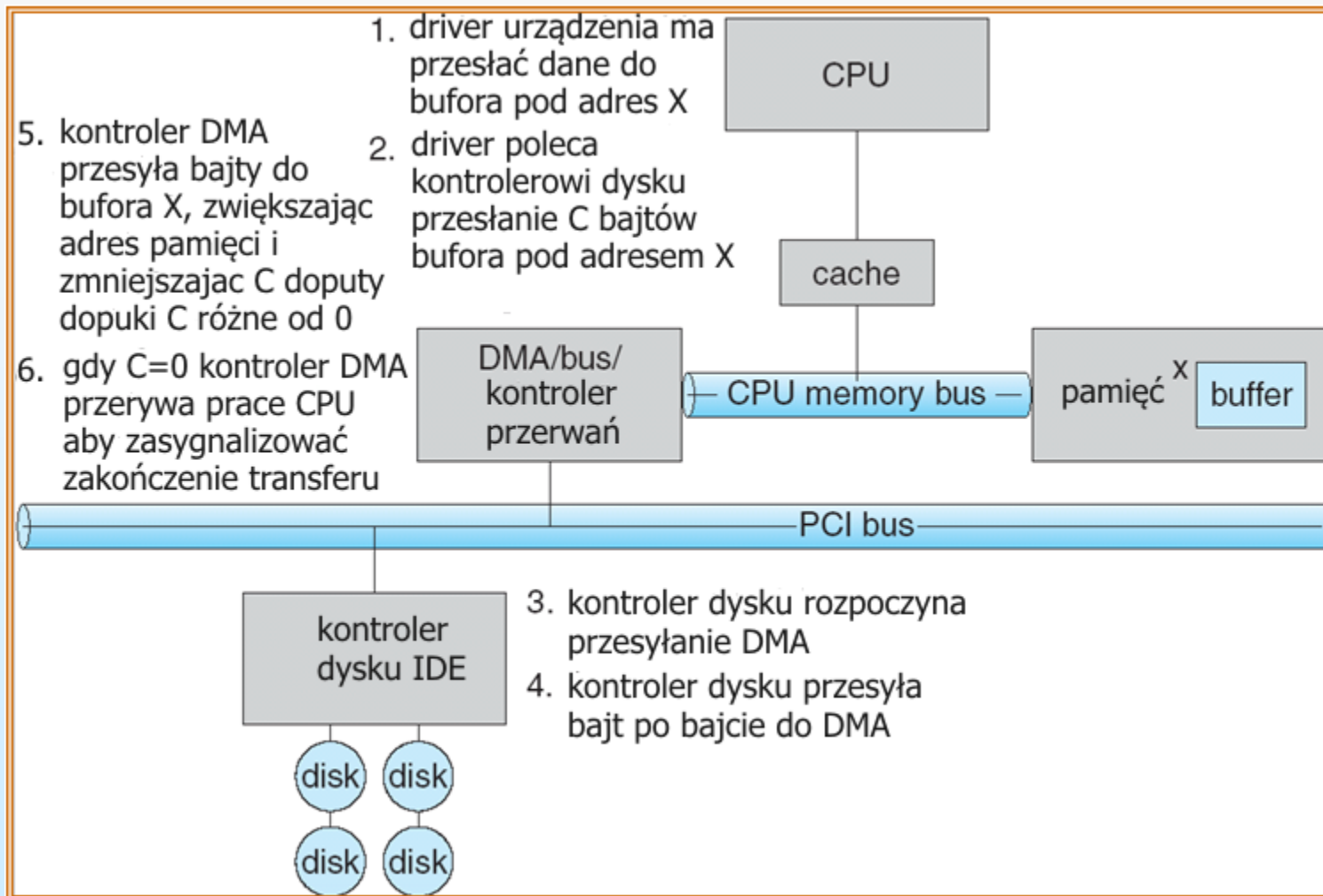
Wektor zdarzeń w procesorze Intel Pentium

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts

Bezpośredni dostęp do pamięci (Direct Memory Access)

- Używany do uniknięcia **programowej** obsługi operacji we/wy przesyłających duże porcje danych
- Wymaga kontrolera **DMA**
- Omija procesor aby przenieść dane bezpośrednio pomiędzy we/wy a pamięcią

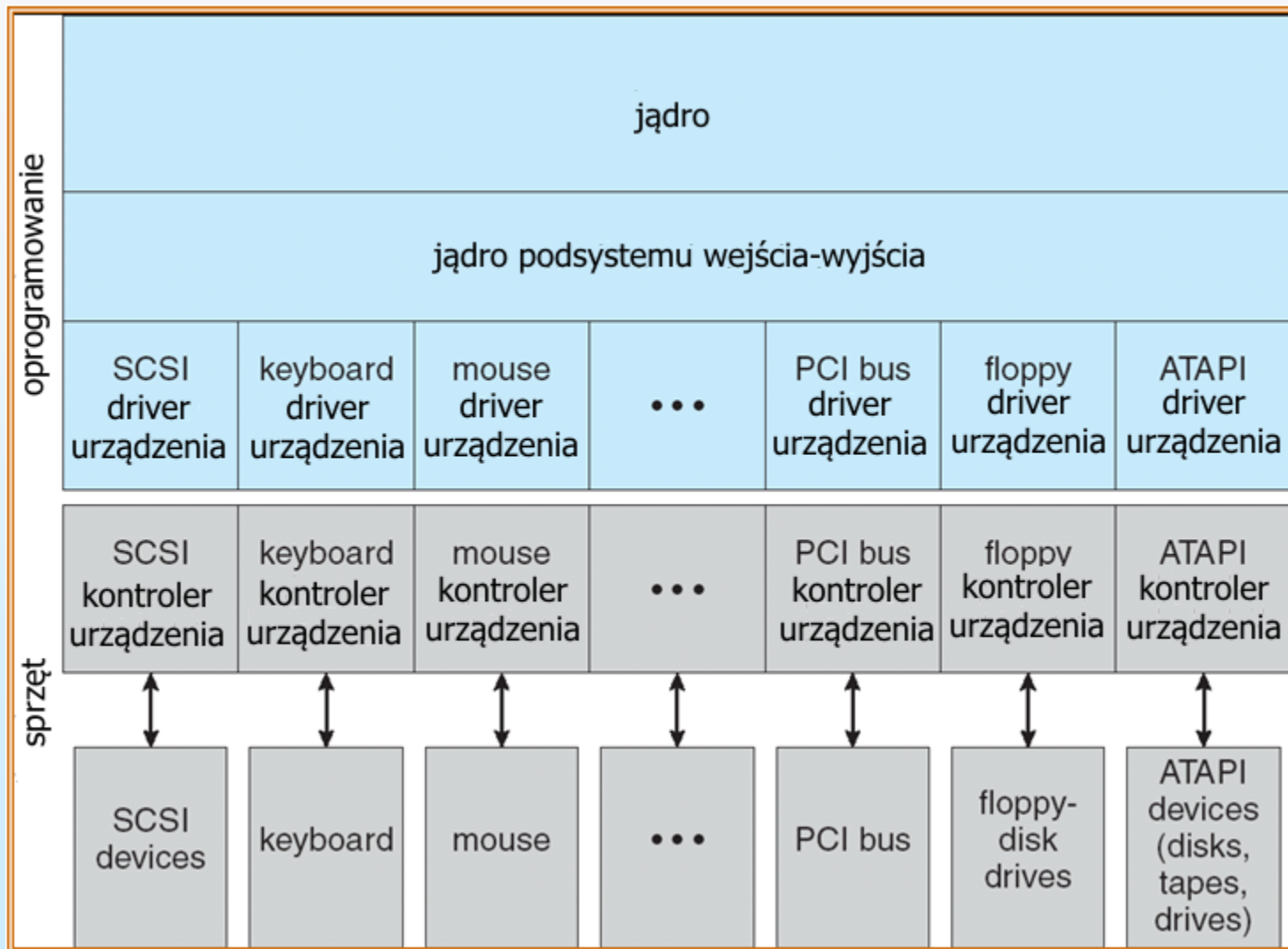
6 kroków procesu przesyłu DMA



Użytkowy (aplikacje) interfejs I/O

- Systemowe wywołania we/wy uogólniają zachowanie [sposób działania] różnych fizycznych urządzeń [enkapsulują]
- Warstwa sterowników urządzeń ukrywa przed jądrem różnice pomiędzy różnymi kontrolerami we/wy
- Urządzenia różnią się w wielu aspektach
 - **Znakowe lub blokowe**
 - **Sekwencyjne lub o swobodnym dostępie**
 - **Dzielone lub dedykowane**
 - **Szybkość operacji**
 - **Zapisująco-odczytujące, tylko do odczytu, tylko do zapisu**

Struktura jądra I/O



Charakterystyka urządzeń I/O

cecha	wariant	przykład
sposób przesyłu danych	znakowy blokowy	terminal dysk
metoda dostępu	sekwencyjny bezpośredni	modem CR-ROM
transfer schedule	synchroniczny asynchroniczny	taśma klawiatura
typ dostępu	dedykowany dzielony	taśma klawiatura
szybkość urządzenia	opóźnienie czas szukania szybkość transmisji opóźnienie między operacjami	
kierunek I/O	tylko odczyt tylko zapis odczyt - zapis	CD-ROM kontroler grafiki dysk

Urządzenia blokowe i znakowe

- Urządzenia blokowe zawierają sterowniki dysków
 - komendy: czytaj, pisz, szukaj
 - surowe we/wy lub dostęp poprzez system plików
 - możliwy dostęp do pliku odwzorowania pamięci
- Urządzenia znakowe to m.in. klawiatury, myszki, porty szeregowo
 - komendy: pobranie, przekazanie (`get` , `put`)
 - biblioteki wyższych warstw umożliwiają edycję całej linii

Urządzenia sieciowe

- Różnią się zdecydowanie od urządzeń blokowych i znakowych poprzez posiadanie własnego interfejsu
- Unix i Windows/NT posiadają interfejs gniazd
 - rozdzielenie protokołów sieciowych od operacji sieciowych
 - umożliwiają funkcjonalność `select`
- Bardzo różne podejścia [filozofie] (potoki, kolejki FIFO, strumienie, kolejki, skrzynki pocztowe)

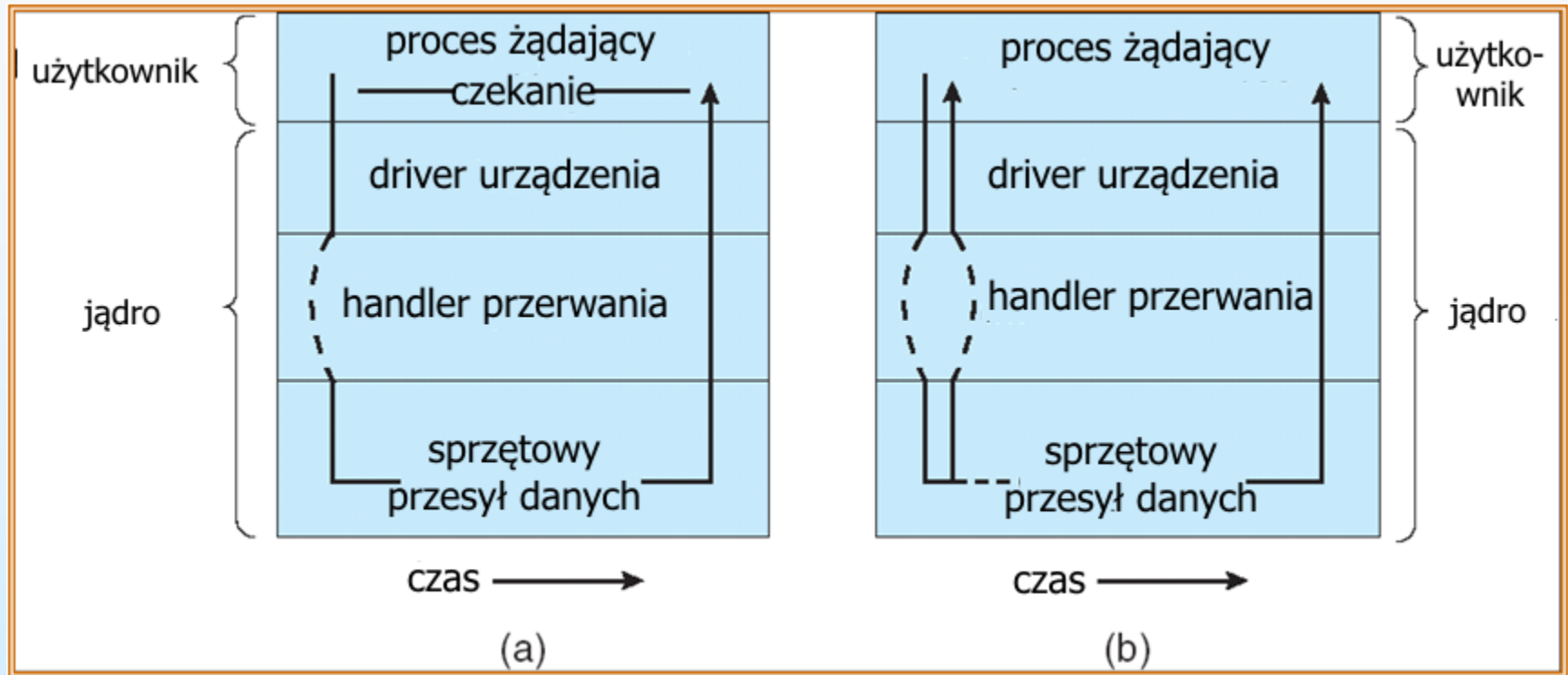
Zegary i Timery

- Podawanie bieżącego czasu, podawanie czasu jaki upłynął ,timer
- **Programowalny timer interwałowy** wykorzystywany do taktowania (timing) , periodycznych przerwań
- `ioctl` (w UNIXie) funkcja umożliwia komunikację na niższym poziomie abstrakcji z urządzeniem np.zegar, timer

I/O z blokowaniem i bez blokowania

- **Blokowanie** – proces jest zawieszony do zakończenia operacji I/O
 - Łatwe w użyciu i do zrozumienia
 - Niewystarczające w pewnych przypadkach potrzeb
- **Bez blokowania** - I/O call returns as much as available
 - Interfejs użytkownika, kopiowanie danych (buforowane I/O)
 - Implementacja poprzez wielowątkowość
 - Szybki powrót z liczbą przeczytanych lub zapisanych bajtów
- **Asynchronicznie** - proces jest wykonywany współbieżnie z operacją we/wy
 - trudny w użyciu
 - podsystem we/wy sygnalizuje [wysyła sygnał] procesowi zakończenie operacji we/wy

Dwie metody I/O



Synchroniczny

Asynchroniczny

Podsystem wejścia-wyjścia w jądrze

- Szeregowanie
 - niektóre żądania wykonania operacji we/wy są porządkowane poprzez zastosowanie kolejek dedykowanych dla każdego urządzenia
 - niektóre systemy operacyjne próbują ...
- Buforowanie – zapamiętywanie danych w pamięci podczas przesyłania pomiędzy urządzeniami
 - rozwiązuje problemy z niedopasowaniem prędkości poszczególnych urządzeń
 - rozwiązuje problemy z niedopasowaniem rozmiaru pakietów danych między poszczególnymi urządzeniami
 - utrzymuje [wspiera] “semantykę kopiowania”

Jądro podsystemu I/O

- **Caching** - szybka pamięć utrzymująca kopię danych
 - Zawsze tylko kopia
 - Kluczowa dla wydajności

- **Spooling** - utrzymywanie danych wyjściowych dla urządzenia
 - stosowane, gdy urządzenie może obsłużyć tylko jedno zadanie naraz
 - np., drukowanie

- **Rezerwacja urządzeń** - umożliwia wyłączny dostęp do urządzenia
 - wywołania systemowe dla alokacji i dealokacji [rezerwacji i zwalniania]
 - możliwość wystąpienia deadlocków

Obsługa błędów

- Systemy operacyjne mogą odzyskiwać dane z dysku, urządzenia niedostępnego i chwilowych błędów zapisu
- Większość operacji we/wy zwraca w wyniku numer lub kod błędu w wypadku niepowodzenia
- Logi systemowe utrzymują raporty o błędach

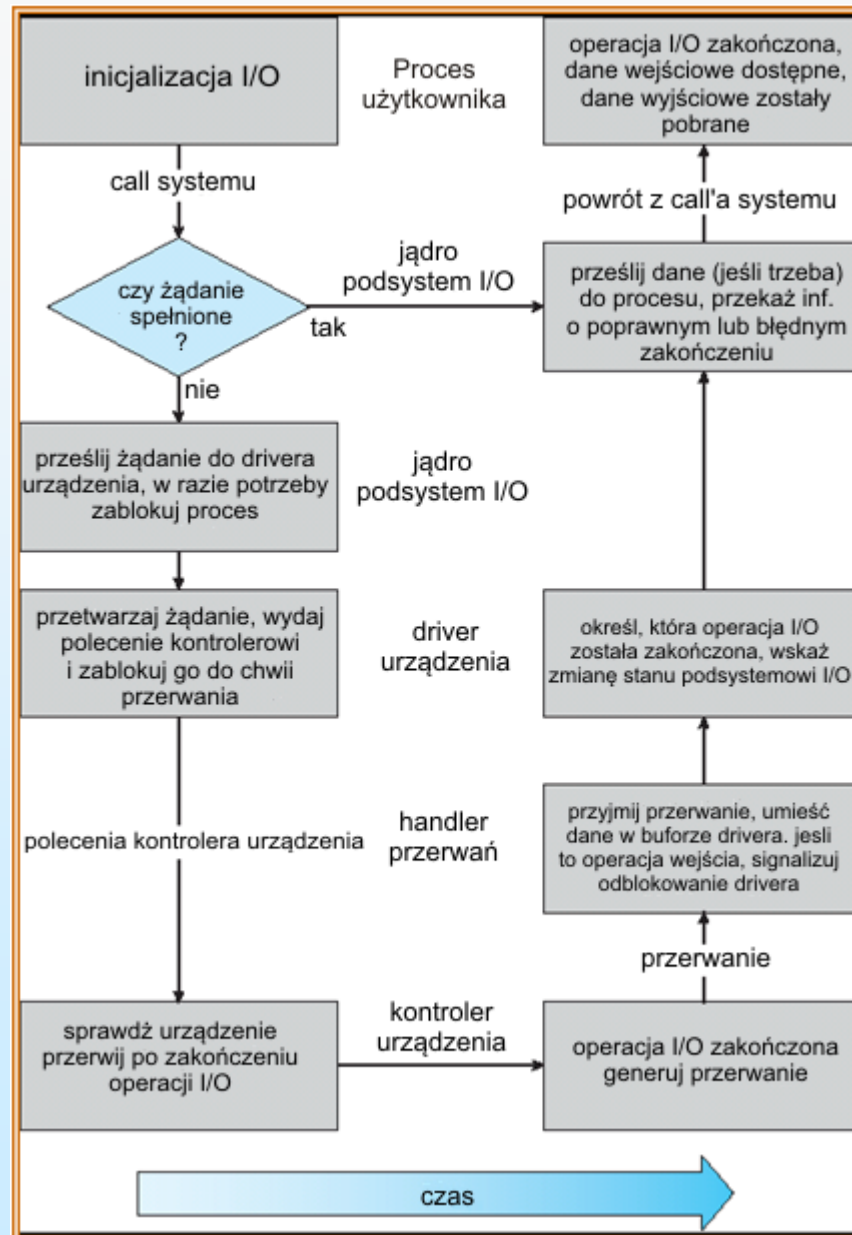
Struktura danych jądra

- Jądro utrzymuje stan informacji o składnikach we/wy, w tym tabel otwartych plików, połączeń sieciowych, stanu urządzeń znakowych
- Wiele złożonych struktur danych dla śledzenia buforów, alokacji pamięci, „brudnych” bloków
- Zdarza się użycie metod obiektowych i przesyłania komunikatów do zaimplementowania we/wy

Przekształcenie zamówień wejścia-wyjścia na operacje sprzętowe

- Analiza odczytu pliku z dysku dla procesu:
 - określ urządzenie, na którym znajduje się plik
 - przetłumacz nazwę pliku na reprezentację sprzętową
 - fizycznie odczytaj dane z dysku do bufora
 - udostępnij dane żądającemu procesowi
 - zwróć sterowanie do procesu

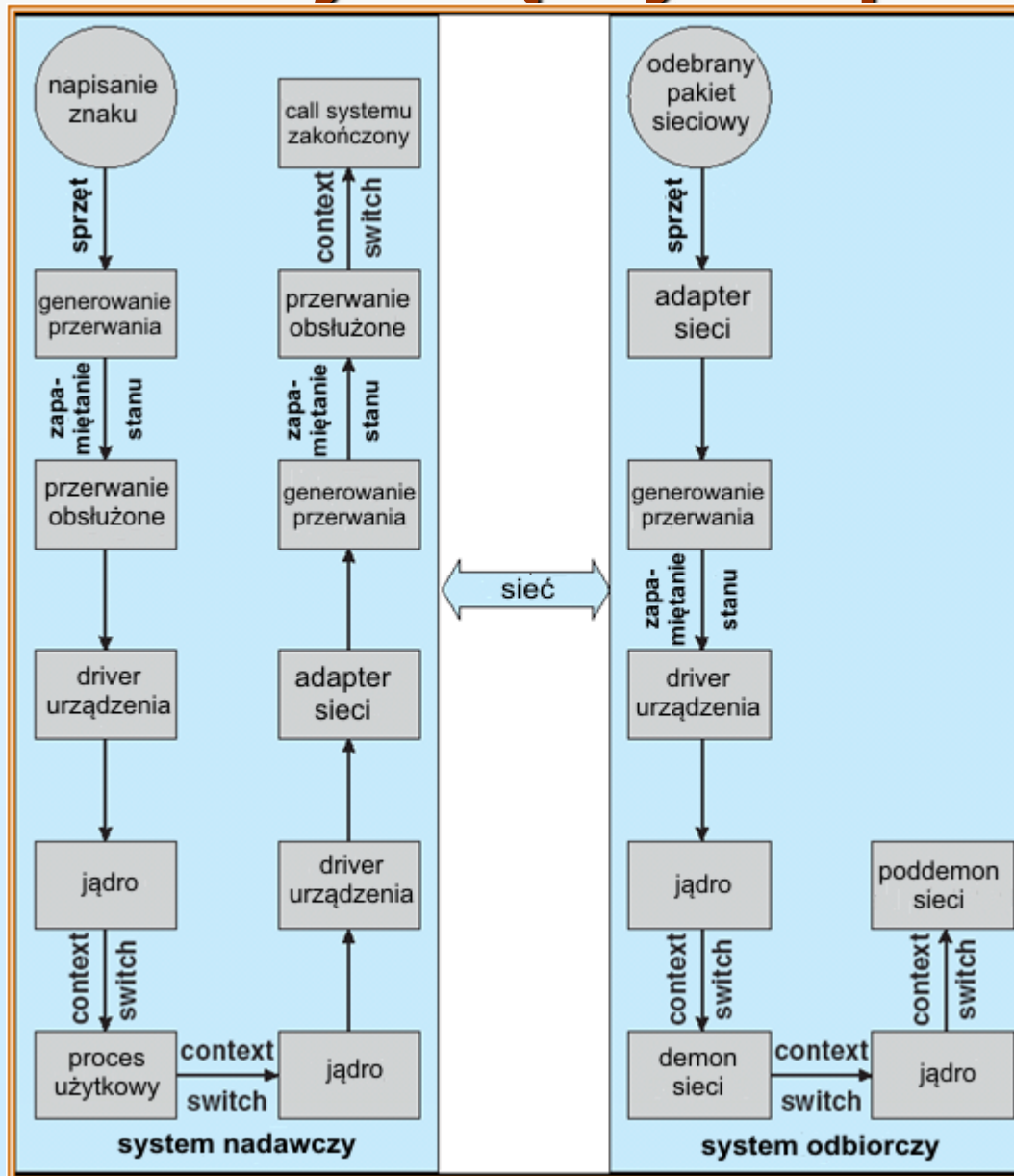
Cykl obsługi żądania I/O



Wydajność

- Operacje we/wy jako podstawowy czynnik wydajności systemu:
 - wymaga zaangażowania procesora do wykonania kodu sterownika i operacji w trybie jądra
 - przełączanie kontekstu dla obsługi przerwania
 - kopiowanie danych
 - ruch sieciowy szczególnie wpływa na wydajność

Komunikacja międzykomputerowa



Poprawa wydajności

- Ogranicz liczbę przełączeń kontekstu
- Ogranicz liczbę kopiowań danych
- Ogranicz przerwania poprzez stosowanie dużych transferów danych, inteligentnych kontrolerów, aktywnych odpytywań
- Użyj DMA
- Wyważ i zbalansuj [dostrój] użycie CPU, pamięci, magistrali i wydajności we/wy dla największej przepustowości

Koniec