

Wykład 8. Systemy wejścia- wyjścia

Jarosław Koźlak

Wprowadzenie

Różne rodzaje urządzeń zewnętrznych:

- dyski twarde
- dyski CD
- myszki
- klawiatury
- ...

Tendencje:

- wzrost standaryzacji interfejsów programowych i sprzętowych
- wzrost różnorodności urządzeń zewnętrznych

Podstawowe elementy systemu wejścia-wyjścia:

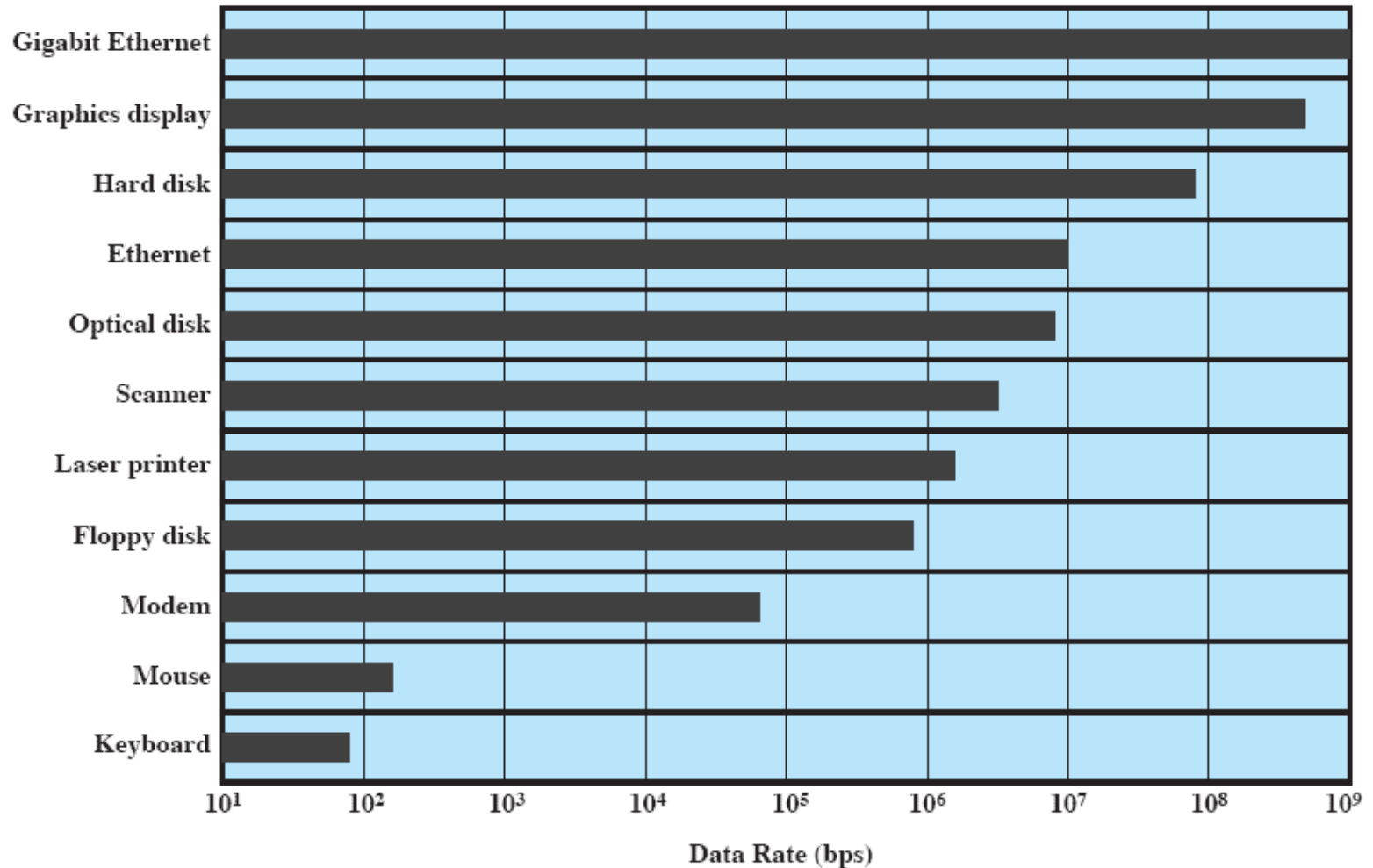
- sprzęt: porty, szyny, sterowniki urządzeń (ang. *device controllers*)
- oprogramowanie: moduły sterujące (ang. *device drivers*) - tworzą jednolity interfejs dostępu do podsystemu wejścia-wyjścia

Podstawowe elementy systemu wejścia-wyjścia

- sprzęt: porty, szyny, sterowniki urządzeń (ang. device controllers)
- oprogramowanie: moduły sterujące (ang. device drivers) - tworzą jednolity interfejs dostępu do podsystemu wejścia-wyjścia

Prędkości transmisji urządzeń

- bardzo istotne różnice prędkości urządzeń



Sprzęt wejścia-wyjścia

- **Rodzaje urządzeń:**

- urządzenia pamięci (dyski, czytniki CD, taśmy,...)
- urządzenia przesyłania danych (karty sieciowe, modemy)
- urządzenia interfejsu z człowiekiem (ekrany monitorów, klawiatury, myszki)

- **Podłączenia urządzenia:**

- port - punkt łączący urządzenie z maszyną
- szyna - wiązka przewodów ze ściśle zdefiniowanym protokołem, który precyzuje zbiór komunikatów, które mogą być przesyłane tymi przewodami

Komunikacja procesora ze sterownikiem

- **Sterownik (ang. controller)** - zespół układów elektronicznych kierujących pracą portu, szyny, lub urządzenia

Sposoby komunikacji procesora ze sterownikiem:

- sterownik dysponuje rejestrami do pamiętania danych i sygnałów sterujących,
- procesor komunikuje się ze sterownikiem odczytując i zapisując bity w tych rejestrach
- realizacja komunikacji:
- zastosowanie specjalnych rozkazów umożliwiających odwoływanie się do rejestrów urządzenia
- sterownik urządzenia umożliwia operacje we/wy odwzorowywane w pamięci operacyjnej - rejestry sterujące urządzeniem są odwzorowywane w przestrzeni adresowej procesora

Przykładowe rejestry portu wejścia/wyjścia

- stan (ang. *status*) - bity czytane przez procesor główny, które określają zakończenie bieżącego polecenia, dostępność bajtu do czytania w rejestrze danych wejściowych, błąd urządzenia
- sterowanie (ang. *control*) - zapisywany przez procesor głównym w celu rozpoczęcia polecenia, zmiany trybu pracy urządzenia
- dane wejściowe (ang. *data in*) - zawiera dane z urządzenia pobierane przez procesor
- dane wyjściowe (ang. *data out*) - zawiera dane wysyłane przez procesor do urządzenia

Komunikacja procesor główny - sterownik

Metoda: Odpytywanie (ang. polling)

Bity specjalnego przeznaczenia

- bit zajętości - stosowany przez sterownik:
 - ustawiony - gdy sterownik jest zajęty pracą
 - wyczyszczony - gdy sterownik jest gotów do przyjęcia następnego polecenia
- bit gotowości polecenia (ang. command ready)
 - ustawiony, gdy polecenie do wykonania jest dostępne dla sterownika

Komunikacja procesor główny - sterownik

Metoda: odpytywanie

- **Przykład: procesor pisze na wyjściu**
1. **Procesor powtarza czytanie bitu zajętości dopóty, dopóki bit ten nie przyjmie wartości 0**
 2. **Procesor główny ustawia bit pisania (ang. write bit) w rejestrze poleceń i wpisuje bajt do rejestru danych wyjściowych**
 3. **Procesor główny ustawia bit gotowości polecenia,**
 4. **Gdy sterownik zauważy że bit gotowości polecenia jest ustawiony, wówczas ustawia bit zajętości**
 5. **Sterownik czyta rejestr poleceń i rozpoznaje polecenie pisania, zatem czyta bajt z rejestru danych wejściowych i wykonuje na urządzeniu operację we/wy**
 6. **Sterownik czyści bit gotowości polecenia, oraz bit błędu w rejestrze stanu, aby powiadomić, że operacja we/wy zakończyła się pomyślnie, po czym czyści buj zajętości, sygnalizując, że zakończył działanie**

powyższa pętla - powtarzana dla każdego bajtu

- **Krok 1 - procesor główny wykonuje aktywne czekanie (ang. busy waiting)/ odpytywanie (ang. polling)**
 - **nieefektywne w przypadku wolnych urządzeń**

Przerwania

- zamiast ciągłego odpytywania sterownika przez procesor, sterownik powiadamia procesor o przejściu do stanu gotowości działania
- procesor jest wyposażony w połączenie zwane linią zgłaszania przerw (ang. interrupt request line), którą bada po wykonaniu każdego rozkazu
- jeśli sterownik sygnalizuje zgłoszenie, to procesor przechowuje dane określające aktualny stan i wykonuje skok do procedury obsługi przerw (ang. interrupt handler)
- procedura obsługi przerw rozpoznaje przyczynę przerw, wykonuje niezbędne czynności i rozkaz powrotu z przerw, który przywraca stan procesora sprzed przerw

Wymagane własności systemu obsługi przerw

- Wymaganie, które powinien spełniać system obsługi przerw (złożony z jednostki centralnej i sterownika przerw, ang. interrupt controller):
 - zapewnienie możliwości opóźnienia obsługi przerwania, jeżeli są wykonywane działania krytyczne
 - zapewnienie skutecznego sposobu kierowania przerwania do właściwej procedury obsługi bez konieczności odpytywania wszystkich urządzeń, które z nich zgłosiło przerwanie
 - przerwanie wielopoziomowe - system operacyjny powinien móc odróżnić rodzaje przerw o wysokim i niskim priorytecie

Przerwania cd.

Typowo: dwie linie zgłaszania przerwań:

- przerwania niemaskowalne (dla zdarzeń, które nie mogą zostać zignorowane np. błędy pamięci)
- przerwania maskowalne, które można wyłączyć przed wykonaniem ciągu instrukcji, który nie powinien zostać przerwany (używane przez sterowniki urządzeń do zgłaszania żądań obsługi)

Wybór procedury obsługi przerwania:

- przekazywany adres służący do wyboru procedury, zazwyczaj pozycja w tablicy - wektorze przerwań (ang. interrupt vector)
- w praktyce komputery mają więcej urządzeń (procedur obsługi przerwań) niż pozycji adresowych w wektorze przerwań:
 - technika łańcuchowania przerwań
 - każdy element wektora przerwań wskazuje na czoło listy procedur obsługi przerwań
 - procedury z listy są wywołane kolejno, aż zostanie odnaleziona ta, która powinna obsłużyć przerwanie

Zastosowania mechanizmu obsługi przerwań

- podczas startu systemu są sprawdzane szyny sprzętowe, określa się, jakie urządzenia są dostępne i instaluje się w wektorze przerwań odpowiednie procedury obsługi
- przerwania generowane podczas operacji we/wy, gdy sterowniki zgłaszają swoją gotowość
- obsługa sytuacji wyjątkowych (dzielenie przez 0, adresowanie chronionego/ nie istniejącego obszaru pamięci)
- podczas stronicowania pamięci wirtualnej - brak strony powoduje zgłoszenie przerwania
- realizacja funkcji systemowej (ang. system call) - przywołanie usługi jądra:
 - sprawdza się podane argumenty,
 - tworzy się strukturę danych, do przeniesienia ich do jądra
 - wykonywany jest specjalny rozkaz - przerwanie programowe (ang. software interrupt) / pułapka (ang. trap)
 - realizacja funkcji systemowej ma stosunkowo niski priorytet

Bezpośredni dostęp do pamięci

- programowane wejście-wyjście (ang. programmed I/O - PIO) - aktywne czekanie procesora i przekazywanie danych do/z sterownika po 1 bajcie - nieefektywne dla transmisji dużych danych
- **bezpośredni dostęp do pamięci (ang. direct memory access - DMA)** - zastosowanie specjalnego sterownika bezpośredniego dostępu do pamięci

Transmisja w trybie DMA

- procesor główny zapisuje w pamięci *blok sterujący DMA*
- (wskaźnik do źródła przesyłania, wskaźnik miejsca docelowego przesyłania, liczba bajtów do przesłania)
- procesor główny zapisuje adres bloku sterującego w sterowniku DMA i przechodzi do kontynuowania innych prac
- uzgadnianie przesyłania między sterownikiem DMA i sterownikiem urządzenia — realizowane przez parę przewodów :
 - zamówienie DMA (ang. DMA request)
 - potwierdzenie DMA (ang. DMA acknowledge)

Bezpośredni dostęp do pamięci - realizacje przesłania danych

- sterownik urządzenia wytwarza sygnał w przewodzie zamówienia DMA, gdy słowo danych jest gotowe do przesłania
- sterownik DMA przejmuje szynę pamięci, aby umieścić potrzebny adres w liniach adresowych pamięci i wytworzyć sygnał w przewodzie potwierdzenia DMA
- sterownik urządzenia odbiera sygnał potwierdzenia DMA, przesyła słowo danych do pamięci i usuwa sygnał zamówienia DMA
- po zakończeniu całego przesyłania sterownik DMA generuje przerwanie na procesorze głównym

Interfejs wejścia-wyjścia – Rodzaje urządzeń

- **strumień znaków/bloki**
 - urządzenie znakowe przesyła bajty osobno, jeden po drugim
 - urządzenie blokowe przesyła jednorazowo blok bajtów
- **dostęp sekwencyjny/swobodny**
 - urządzenie o dostępie sekwencyjnym przesyła dane po kolei
 - urządzenia o dostępie swobodnym może udostępniać użytkownikowi dane o wybranym przez niego miejscu przechowywania
- **synchroniczność/asynchroniczność**
 - urządzenie synchroniczne - przesyła dane w przewidywalnym z góry czasie
 - urządzenie asynchroniczne - ma nieregularne/nieprzewidywalne czasy odpowiedzi
- **dzielenie lub wyłączność**
 - urządzenie dzielone może być używane współbieżnie przez wiele procesów lub wątków
- **szybkość działania**
 - duża rozpiętość
- **czytanie i pisanie/tylko czytanie/tylko pisanie**
 - przepływ danych w jednym, lub w obu kierunkach

Podsystem wejścia-wyjścia w jądrze

- **Planowanie wejścia-wyjścia**
 - efektywny dobór kolejności wykonania zbioru zamówień
- **Buforowanie**
 - bufor - obszar pamięci,
 - może zawierać dane przesyłane między urządzeniami, lub między urządzeniem i aplikacją
- **Przyczyny buforowania:**
 - rozwiązywanie problemów związanych z dysproporcją między szybkościami producenta i konsumenta danych
 - dopasowanie urządzeń o różnych rozmiarach przesyłanych jednostek danych (np. fragmentowanie/składanie komunikatów przesyłanych przez sieć)
 - zapewnienie semantyki kopii (wersja danych zapisana na dysku jest wersją z chwili odwołania się przez aplikację do systemu) na wejściu i wyjściu aplikacji

Przechowywanie podręczne

- **Spooling i rezerwowanie urządzeń**
 - użycie bufora do przechowywanie danych przeznaczonych dla urządzenia , które nie dopuszcza przeplatania danych w przeznaczonym dla niego strumieniu

Przekształcanie zamówień wejścia-wyjścia na operacje sprzętowe

- Przykład: czytanie pliku z dysku
- program użytkowy odwołuje się do danych za pomocą nazwy pliku
- system plików zajmuje się odwzorowaniem nazwy plików za pomocą katalogów systemu plików i uzyskuje dostęp do miejsca na dysku przydzielonym plikowi
 - w systemie MS DOS nazwa jest odwzorowywana na liczbę wskazującą pozycję w tablicy FAT
 - w systemie UNIX nazwa jest odwzorowywana na numer i-węzła, który zawiera informacje o przydziale miejsca na dysku

Powiązanie nazwy pliku ze sterownikiem dysku

- **System MS DOS**

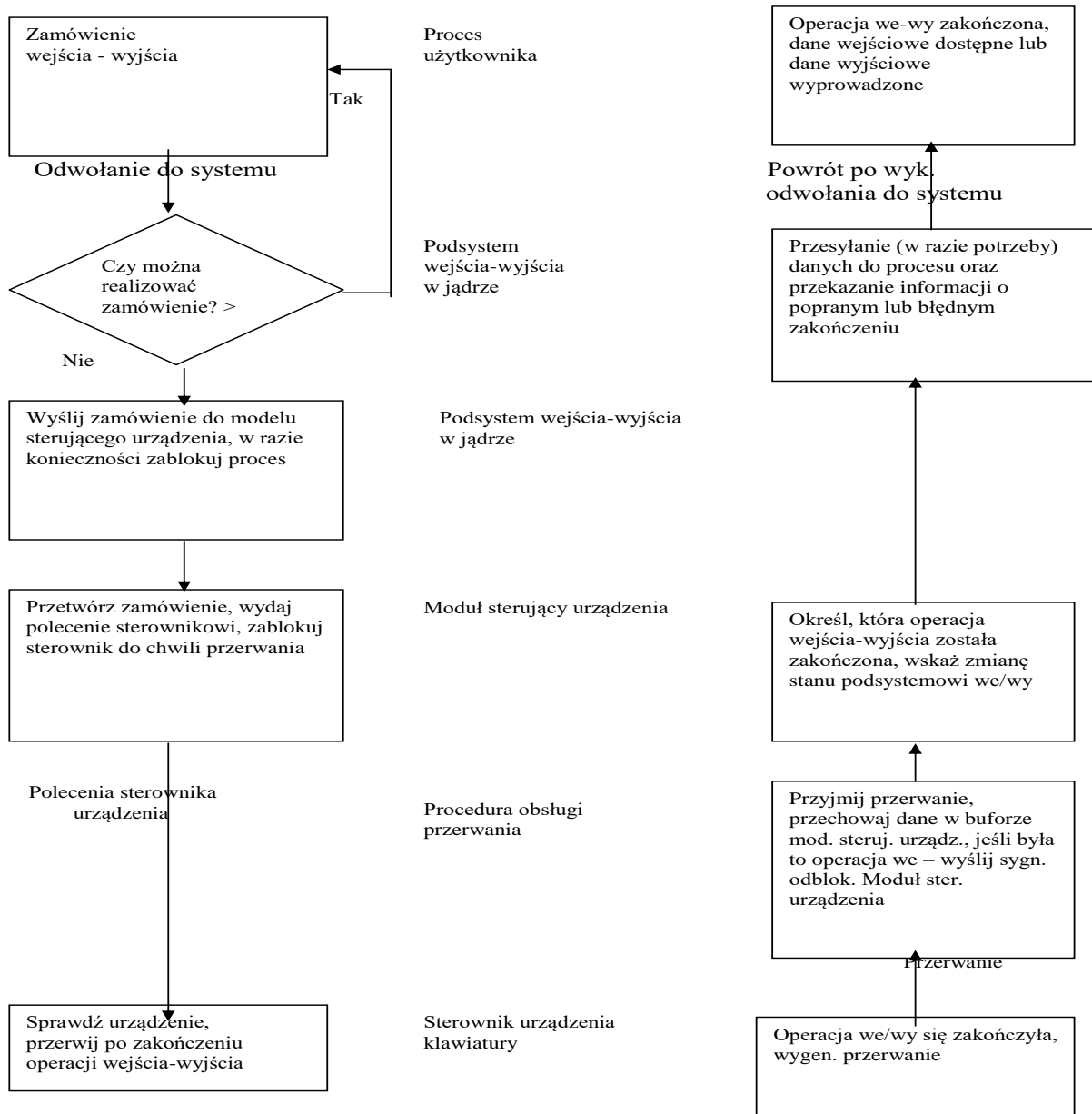
- początek nazwy pliku (do znaku „:”) identyfikuje jednostkę sprzętu, w tablicy urządzeń określa odpowiedni adres portu
- pozostała część nazwy pliku – nazwa pliku w obrębie urządzenia

Powiązanie nazwy pliku ze sterownikiem dysku

System Unix

- przestrzeń nazw urządzeń zaliczona do przestrzeni regularnych nazw systemu plików
- nazwa ścieżki nie wyróżnia części dotyczącej urządzenia
- Unix posiada tablicę montażu, która wiąże przedrostki nazw ścieżek z nazwami poszczególnych urządzeń
 - zamiast numeru i-węzła odnaleziony jest numer urządzenia opisywany poprzez parę <starszy, młodszy>
 - starszy numer urządzenia identyfikuje *moduł sterujący*, który należy wywołać w celu obsługi we/wy urządzenia
 - młodszy numer urządzenia jest przekazywany do modułu sterującego, gdzie służy jako indeks w *tablicy urządzeń*
 - wpis w tablicy urządzeń zawiera odpowiedni adres portu, lub adres odwzorowany w pamięci adres sterownika urządzenia

Etapy wykonania zamówienia wejścia-wyjścia (z blok.)



Etapy realizacji zamówienia operacji wejścia- wyjścia (czytanie z blokowaniem)

1. Proces zamawia w systemie blokującą go operację czytania, powołując się na deskryptor uprzednio otwartego pliku
2. Sprawdzenie poprawności parametrów w kodzie należącej do jądra funkcji systemowej. Jeżeli zamawiana operacja dotyczy we i potrzebne dane znajdują się w buforze pamięci podręcznej, to następuje przekazanie ich do procesu i zamówienie zostaje zrealizowane
3. W przeciwnym wypadku – konieczność wykonania fizycznej operacji we/wy. Proces zostaje usunięty z kolejki procesów gotowych do wykonywania i umieszczony w kolejce czekającej na dane z urządzenia. Podsystem wejścia-wyjścia wysyła zamówienie do modułu sterującego. W zależności od systemu operacyjnego, zamówienie zostaje przesłane za pomocą wywołania podprogramu, lub przez wewnętrzny komunikat jądra
4. Moduł sterujący rezerwuje miejsce na przyjęcie danych w buforze jądra i planuje operację we/wy. W odpowiedniej chwili moduł sterujący wysyła polecenia do sterownika, zapisując je w rejestrach sterujących urządzenia
5. Sterownik urządzenia nakazuje sprzętowym podzespołom urządzenia dokonanie przesłania danych

Etapy realizacji zamówienia operacji wejścia- wyjścia (czytanie z blokowaniem) 2

- ...
- 6. Moduł sterujący może odpytywać o stan urządzenia i dane lub może zorganizować przesłanie w trybie DMA do pamięci jądra. Zakładamy, że przesłanie to jest obsługiwane przez sterownik bezpośredniego dostępu do pamięci, który po zakończeniu przesłania powoduje przerwanie
- 7. Właściwa procedura obsługi przerwania odbiera przerwanie za pośrednictwem tablicy wektorów przerw, zapamiętuje niezbędne dane, przekazuje sygnał do modułu sterującego i kończy obsługę przerwania
- 8. Moduł sterujący urządzenia odbiera sygnał, określa, która operacja we/wy się skończyła, określa stan zamówienia i powiadamia podsystem we/wy w jądrze, że zamówienie zostało zakończone
- 9. Jądro przesyła dane lub przekazuje umowne kody do przestrzeni adresowej procesu, który złożył zamówienie, i przemieszcza proces z kolejki procesów czekających z powrotem do kolejki procesów gotowych do działania
- 10. Przeniesienie procesu do kolejki procesów gotowych powoduje jego odblokowanie. Planista przydziela procesowi procesor, następuje wznowienie pracy procesu po zakończeniu odwołania do systemu

Wydajność: Funkcje we/wy mogą być implementowane:

- w oprogramowaniu aplikacji
- w module sterującym
- w sprzęcie



wzrost wydajności

wzrost elastyczności

Jak można poprawiać wydajność we/wy:

- zmniejszyć liczbę przełączeń kontekstu
- zmniejszyć liczbę kopiowań danych w pamięci
- zmniejszyć częstość występowania przerwań przez stosowanie wielkich przesłań, lub odpytywania
- zwiększyć współbieżność za pomocą sterowników pracujących w trybie DM lub kanałów, w celu odciążenia procesora głównego
- realizować elementarne działania za pomocą sprzętu i pozwalać na ich współbieżne wykonanie w sterownikach urządzeń, przy jednoczesnym działaniu szyny i procesora
- równoważyć wydajność procesora, podsystemów pamięci, szyny i operacji we/wy, ponieważ przeciążenie w jednym miejscu będzie powodować bezczynność w innych miejscach

Planowanie dostępu do dysku

- usługi dyskowe dają duży narzut czasowy dla aplikacji,
- zamówienia do dysku (jak każdego innego urządzenia we/wy) są kolejgowane,
- SO może skrócić czas obsługi dysku dzięki planowaniu zamówień ,

Budowa dysku

- adresowane jako wielkie jednowymiarowe tablice bloków logicznych, rozmiar bloku logicznego zazwyczaj wynosi 512 B
- tablica bloków logicznych jest odwzorowywana na sektory dysku,
 - sektor 0 - pierwszy sektor na pierwszej ścieżce najbardziej zewnętrznego cylindra
 - odwzorowanie odbywa się wzdłuż ścieżki, potem pozostałych ścieżek cylindra, a potem w głąb następnych cylindrów - od wewnętrznych do zewnętrznych

Czynniki, od których zależy prędkość dysku:

- przesunięcie głowicy do odpowiedniej ścieżki lub cylindra (trwa czas przeszukania)
- gdy głowica czytająco-pisząca znajduje się nad odpowiednią ścieżką - musi poczekać aż przemieści się pod nią potrzebny blok danych, (czas oczekiwania - latency time)
- przesyłanie danych między dyskiem a pamięcią główną (czas przesyłania)

**Sumaryczny czas obsługi zamówienia dyskowego
: suma trzech powyższych czasów:**

Zamówienie musi obejmować następujące informacje:

- operacja we czy wy,
- adres dyskowy (napęd, cylinder, powierzchnia, blok)
- adres w pamięci głównej,
- wielkość informacji do przesłania

Planowanie dostępu do dysku

- **Planowanie metodą FCFS.**
 - pierwszy nadszedł - pierwszy obsłużony, prosty ale średni czas obsługi może nie być najlepszy
- **Planowanie metodą SSTF (najpierw najkrótszy czas przeszukiwania shortest seek-time first)**
 - łączenie obsługi wszystkich zamówień sąsiadujących z bieżącym położeniem głowicy,
- **Planowanie metodą SCAN.**
 - głowica startuje od jednego końca dysku i przemieszcza się do przeciwległego końca, potem zmienia kierunek ruchu itd.
 - Rozważamy skrajne położenie głowicy : największe zagęszczenie nie zrealizowanych zadań dotyczy przeciwległego końca dysku.
- **Planowanie metodą C-SCAN.**
 - głowica przemieszcza się od jednego końca dysku do drugiego obsługując napotkane zamówienia,
 - gdy osiągnie przeciwległy koniec - natychmiast wraca do początku dysku
- **Planowanie metodą LOOK, C-LOOK.**
 - głowica przemieszcza się nie pomiędzy krańcami dysku, ale skrajnymi zamówieniami,

Planowanie dostępu do dysku (2)

- **Wybór systemu planowania:**
 - SCAN, C-SCAN - odpowiednie w systemach z dużą ilością zamówień na operacje dyskowe,
 - SSTF - powszechnie stosowane

RAID

Wprowadzenie

- Wydajność pamięci pomocniczej zwiększa się wolniej niż wydajność procesorów i pamięci głównej
- Dlatego system pamięci dyskowej jest elementem, na który kładziony jest największy nacisk podczas prób zwiększenia ogólnej wydajności systemu komputerowego
- Podobnie jak w przypadku innych komponentów, wzrost wydajności można uzyskać dzięki zastosowaniu wielu równoległych komponentów
 - W przypadku dysków spowodowało to powstanie macierzy dyskowych, pracujących niezależnie i równolegle
 - Jeśli dysków jest kilka,
 - żądania we/wy mogą być obsługiwane równolegle, jeśli dane są zlokalizowane na różnych dyskach
 - pojedyncze żądanie może być zrealizowane równolegle, jeśli dane znajdują się na różnych dyskach
 - Korzystając z wielu dysków można zorganizować dane na wiele sposobów i dodać redundancję, aby zwiększyć niezawodność
 - Mogłoby to utrudniać użycie systemów używanych na różnych platformach, dlatego opracowano standard RAID (ang. Redundant Array of Independent Disks <- nadmiarowa macierz niezależnych dysków)

Poziomy systemu RAID

- System RAID składa się z 7 poziomów - od 0 do 6 (są definiowane inne poziomy, ale nie są powszechnie uznane)
- Poziomy nie sugerują hierarchicznej zależności, ale wyznaczają odmienne architektury, które mają trzy wspólne cechy:
 - RAID to zbiór dysków fizycznych postrzegany przez system operacyjny jako jeden dysk
 - Dane są rozproszone po fizycznych dyskach wchodzących w skład macierzy
 - Nadmiarowa pojemność dysku służy do przechowywania informacji o parzystości, co pozwala odtworzyć dane w przypadku awarii dysku (ta cecha zależy od poziomu RAID, RAID 0 nie ma tej właściwości)

Poziomy RAID

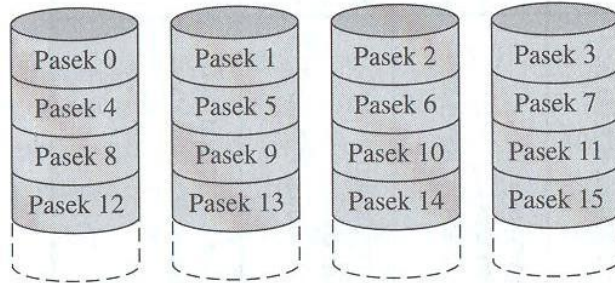
Kategoria	Pozio m	Opis	Tempo żądań we- wy (zapis- odczyt)	Szybkość transmisji danych	Typowe zastosowanie
Paskowanie	0	Brak nadmiarowości	Duże paski: doskonałe	Małe paski: doskonała	Aplikacje wymagające wysokiej jakości przetwarzania niekrytycznych danych
Kopiowanie lustrzane	1	Kopia lustrzana	Dobre/średnie	Średnia/ średnia	Dyski systemowe, pliki krytyczne
Dostęp równoległy	2	Nadmiarowość z wykorzystaniem kodu Hamminga	Niskie	Doskonała	
	3	Parzystość z przeplotem bitów	Niskie	Doskonała	Aplikacje z dużymi żadaniami we-wy, np. przetwarzanie obrazów, CAD
Dostęp niezależny	4	Parzystość z przeplotem bloków	Doskonałe/ średnie	Średnia/ Niska	
	5	Rozproszona parzystość z przeplotem bloków	Doskonałe/ średnie	Średnia/ Niska	Duże tempo żądań, dużo operacji odczytu, wyszukiwanie danych
	6	Podwójnie rozproszona parzystość z przeplotem bloków	Doskonałe/ niskie	Średnia/ niska	Aplikacje wymagające bardzo wysokiej niezawodności

Poziomy macierzy RAID

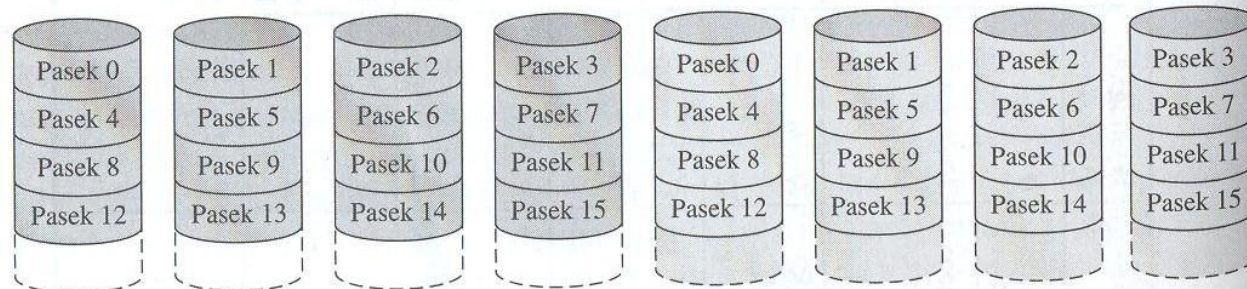
Tabela 11.4. Poziomy macierzy RAID

Kategoria	Poziom	Opis	Wymagane dyski	Dostępność danych	Duża przepustowość operacji we/wy	Niewielki współczynnik żądań we/wy
Paskowanie	0	Bez nadmiarowości	N	Niższa niż w przypadku jednego dysku	Bardzo duża	Bardzo wysoki zarówno dla odczytu, jak i zapisu
Kopiowanie lustrzane	1	Kopia lustrzana	$2N$, $3N$ itd.	Wyższa niż w RAID 2, 3, 4 lub 5, niższa niż w RAID 6	Wyższa niż odczyt z jednego dysku, podobna przy zapisie na jednym dysku	Do dwóch razy wyższy dla odczytu z pojedynczego dysku; podobny dla zapisu
Dostęp równoległy	2	Nadmiarowość z wykorzystaniem kodu Hamminga	$N + m$	O wiele wyższa niż w jednym dysku, wyższa niż RAID 3, 4 lub 5	Najwyższa ze wszystkich wymienionych alternatyw	W przybliżeniu dwukrotnie wyższy od jednego dysku
	3	Parzystość z przeplotem bitów	$N + 1$	O wiele wyższa niż w jednym dysku, porównywalna z RAID 2, 4 lub 5	Najwyższa ze wszystkich wymienionych alternatyw	W przybliżeniu dwukrotnie wyższy od jednego dysku
Dostęp niezależny	4	Parzystość z przeplotem bloków	$N + 1$	O wiele wyższa niż w jednym dysku, porównywalna z RAID 2, 3 lub 5	Podobna do RAID 0 dla odczytu, znacząco niższa dla zapisu na jednym dysku	Podobny do RAID 0 dla odczytu, znacząco niższy dla zapisu na jednym dysku
	5	Parzystość rozproszona z przeplotem bloków	$N + 1$	O wiele wyższa niż w jednym dysku, porównywalna z RAID 2, 3 lub 4	Podobna do RAID 0 dla odczytu, mniejsza niż w jednym dysku dla zapisu	Podobny do RAID 0 dla odczytu, ogólnie niższy niż w jednym dysku dla zapisu
	6	Podwójnie rozproszona parzystość z przeplotem bloków	$N + 2$	Najwyższa ze wszystkich wymienionych alternatyw	Podobna do RAID 0 dla odczytu, niższa niż RAID 5 dla zapisu	Podobny do RAID 0 dla odczytu, znacząco niższy niż RAID 5 dla zapisu

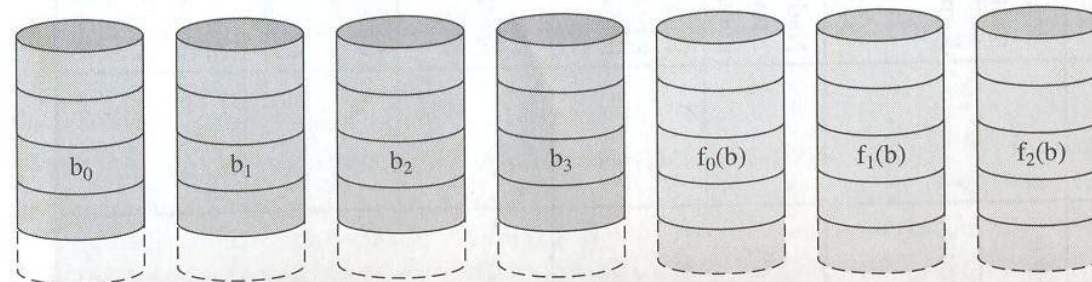
RAID 0, RAID1, RAID2



(a) RAID 0 (bez nadmiarowości)

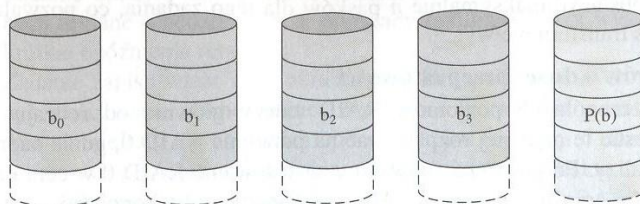


(b) RAID 1 (kopia lustrzana)

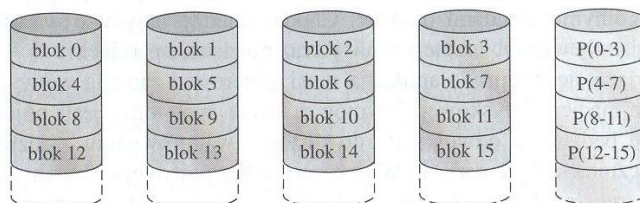


RAID 3-6

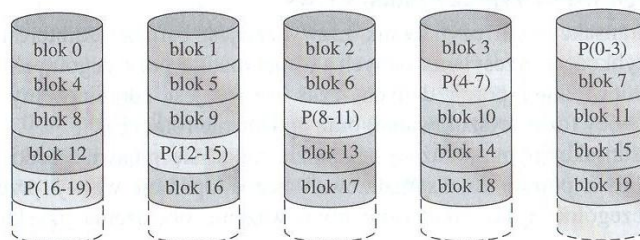
rzy, nosi nazwę *pasma*.



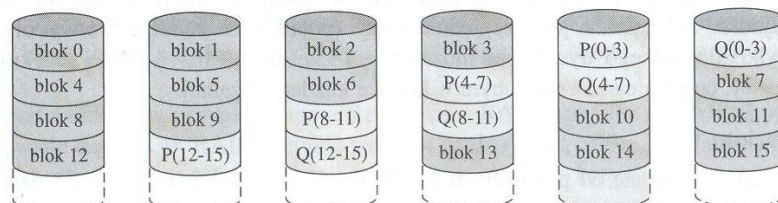
(d) RAID 3 (parzystość z przeplotem bitów)



(e) RAID 4 (parzystość z przeplotem bloków)



(f) RAID 5 (parzystość rozproszona z przeplotem bloków)



(g) RAID 6 (podwójna nadmiarowość)

RAID-O

- Nie zapewnia nadmiarowości,
- W niektórych sytuacjach podstawową kwestią jest wydajność, pojemność i niski koszt
- Dane użytkownika i systemu są rozproszone po wszystkich dyskach macierzy
 - Korzyści: jeśli zgłoszone zostaną dwa żądania dostępu do dwóch różnych bloków danych, jest duża szansa, że są one na różnych dyskach, a zatem zlecenia mogą być realizowane równolegle
- Dane są nie tylko rozproszone po dyskach, ale także są rozciągnięte w paski (ang. strips) na wszystkich dostępnych dyskach
 - Wszystkie dane znajdują się pozornie na jednym dostępnym dysku
 - Dysk jest podzielony na paski, paskami mogą być fizyczne bloki, sektory albo inne jednostki
 - Paski są cyklicznie odwzorowywane na kolejne elementy macierzy
 - W macierz n-dyskowej pierwszych n logicznych pasków przechowuje się fizycznie w pierwszych paskach na każdym z n dysków (co tworzy pierwsze pasmo), kolejnych n logicznych pasków przechowuje się fizycznie w drugich paskach itd.
 - Zaleta: jeśli jedno żądanie we-wy dotyczy wielu logicznie ciągłych pasków, to n pasków można obsłużyć równolegle, bardzo skracając czas transmisji danych
 - Oprogramowanie zarządzające macierzą odwzorowuje logiczną przestrzeń adresową na przestrzeń fizyczną
- Korzyści
 - Zapewnia szybka transmisja danych
 - Wymogi: ścieżka między pamięcią a dyskami ma dużą przepustowość,
 - Aplikacja musi zgłaszać żądania efektywnie sterujące macierzą (żądania dotyczą dużej – w porównaniu z wielkością paska – ilości danych)
 - Umożliwia szybką realizację wielu żądań we-wy (rozkład obciążenia na wiele dysków)

RAID-1

- Inny sposób uzyskania nadmiarowości niż w przypadkach 2-6
- W innych przypadkach nadmiarowość uzyskuje się dzięki obliczaniu parzystości, natomiast w przypadku RAID 1 powiela się wszystkie dane
- Dane są na paski jak w RAID 0, ale każdy pasek logiczny jest odwzorowany na dwa odrębne dyski fizyczne
- Zalety:
 - Żądanie odczytu może być obsłużone przez dowolny z dwóch dysków zawierających żądane dane, można wybrać więc ten z krótszym czasem oczekiwania i opóźnieniem rotacyjnym
 - Żądanie zapisu wymaga aktualizacji obu odpowiadających sobie pasków, ale można to zrobić równolegle, zatem wydajność zapisu jest uwarunkowana przez wolniejszy z dwóch zapisów (większy czas wyszukiwania i opóźnienie rotacyjne). Nie występuje pogorszenie wydajności zapisu danych (w RAID 2-6 używa się bitów parzystości, podczas aktualizacji jednego paska trzeba także zaktualizować bity parzystości)
 - Usuwanie skutków jest bardzo proste. Kiedy dysk ulegnie awarii, dane można odczytać z drugiego dysku.
- Wady:
 - Koszt – trzeba dwukrotnie większej przestrzeni dyskowej niż rozmiar obsługiwanego dysku logicznego
 - Dlatego konfiguracje RAID są zwykle ograniczone do dysków przechowujących oprogramowanie, dane systemowe i inne krytyczne pliki
- W środowisku transakcyjnym może zapewnić dużą szybkość żądań we-wy, jeśli większość żądań dotyczy odczytu danych (niemal dwukrotnie większa niż RAID 0)
- Jeśli duża część dotyczy zapisu danych, to prędkość nie będzie istotnie mniejsza

RAID-2

- Poziomu RAID-2 i RAID-3 wykorzystują technikę dostępu równoległego
- W macierzy o dostępie równoległym, w realizacji każdego żądania uczestniczą wszystkie dyski
- Zwykle poszczególne dyski są zsynchronizowane w taki sposób, że w danym momencie każda głowica znajduje się w tym samym położeniu nad każdym z dysków
- Podobnie jak w innych odmianach RAID stosuje się paskowanie danych (ang. data stripping)
- W przypadku RAID 2 i RAID 3 paski są bardzo małe, często rozmiaru jednego bitu lub słowa
- Na poziomie RAID 2 kod korelacji danych oblicza się na podstawie odpowiadających sobie bitów na każdym dysku danych, a bity kodu są przechowywane w odpowiednich pozycjach bitowych na wielu dyskach parzystości. Zwykle używa się kodu Hamminga, który potrafi naprawić błędy jednobitowe i wykryć dwubitowe,
- Poziom RAID 2 wymaga mniejszej liczby dysków niż RAID 1, ale nadal jest kosztowny (ilość nadmiarowych dysków – proporcjonalna do logarytmu liczby dysków danych)
- Macierz RAID 2 byłaby efektywna tylko w środowisku, w którym występuje wiele błędów dyskowych, ze względu na wysoką niezawodność pojedynczych dysków poziom RAID 2 jako przesadny nie jest implementowany.

RAID-3

- Zorganizowany podobnie jak RAID 2, wymaga jednak tylko jednego nadmiarowego dysku bez względu na rozmiar macierzy
- RAID 3 stosuje dostęp równoległy, z danymi podzielonymi na małe paski
- Zamiast kodu korekcji błędu oblicza się prosty bit parzystości dla zbioru pojedynczych bitów znajdujących się na tej samej pozycji na wszystkich dyskach danych
- Nadmiarowość
 - W przypadku awarii dysku odczytuje się dysk parzystości i rekonstruuje dane na podstawie zawartości pozostałych urządzeń
 - Po wymianie uszkodzonego dysku można odtworzyć brakujące dane na nowym dysku i wznowić działanie macierzy
 - W razie awarii dysku wszystkie dane są dostępne w tzw. trybie zredukowanym.
 - W przypadku odczytów brakujące dane są odtwarzane w locie poprzez obliczenie różnicy symetrycznej (XOR)
 - Podczas zapisu danych w zredukowanej macierzy RAID 3 trzeba utrzymać spójność parzystości w celu późniejszego odtworzenia danych
 - Powrót do zwykłego trybu pracy wymaga wymiany uszkodzonego dysku i odtworzenia całej jego zawartości na nowym dysku
- Wydajność
 - Dane są podzielone na małe paski, RAID 3 pozwala uzyskać bardzo wysoką prędkość transmisji. Każde żądanie we-wy wiąże się z równoległą transmisją danych ze wszystkich dysków danych.
 - Zwiększenie wydajności jest szczególnie widoczne w przypadku dużych transmisji.
 - Można realizować tylko jedno żądanie we-wy jednocześnie, więc w środowisku transakcyjnym wydajność spada

RAID-4

- Poziomy od RAID 4 do RAID 6 wykorzystują technikę niezależnego dostępu.
- W macierzy o niezależnym dostępie każdy składowy dysk pracuje niezależnie, co pozwala na równoległą realizację żądań we-wy.
- Macierze o niezależnym dostępie sprawdzają się lepiej w zastosowaniach, które wymagają dużego tempa obsługi żądań we-wy, a gorzej tam, gdzie wymagana jest transmisja danych
- Stosuje się paskowanie danych
 - Na poziomach 4-6 paski są względnie duże
 - W RAID 4 pasek parzystości oblicza się bit po bicie na podstawie odpowiednich pasków na wszystkich dyskach danych, bity parzystości przechowuje się na odpowiednim pasku na dysku parzystości
- W RAID 4 występuje pogorszenie wydajności podczas zapisu małej ilości danych.
 - Podczas każdego zapisu oprogramowanie zarządzające macierzą musi zaktualizować nie tylko dane użytkownika, ale także odpowiednie bity parzystości
- Wydajność:
 - Przy zapisie, aby obliczyć nową parzystość, oprogramowanie zarządzające macierzą musi odczytać stary pasek użytkownika i stary pasek parzystości, a następnie zaktualizować te dwa paski nowymi danymi i nowo obliczoną parzystością.
 - Zatem każdy zapis paska wiąże się z dwoma odczytami i dwoma zapisami
 - W przypadku większego zapisu we-wy, który obejmuje paski na wszystkich dyskach, można obliczyć parzystość na podstawie nowych danych. Dysk parzystości można uaktualniać równolegle z dyskami danych, nie występują wtedy dodatkowe operacje odczytu i zapisu
 - Każda operacja zapisu angażuje dysk parzystości, który przez to może stać się wąskim gardłem

RAID-5

- Macierz RAID 5 jest zorganizowana podobnie do RAID 4
- Różnica polega na tym, że w RAID 5 paski parzystości są rozproszone na wszystkich dyskach
- Przydział pasków danych jest zwykle cykliczny, w przypadku macierzy n-dyskowej pasek parzystości znajduje się na innym dysku dla pierwszych n pasków, a następnie wzór się powtarza
- Rozłożenie pasków parzystości na wszystkich dyskach pozwala uniknąć potencjalnego gardła we-wy, które może wystąpić w macierzy RAID

RAID-6

- W RAID 6 przeprowadza się dwa różne obliczenia parzystości i zapisuje ich wyniki w oddzielnych blokach na różnych dyskach
- Zatem macierz RAID, w której dane użytkownika zajmują n dysków, składa się z $n+2$ dysków
- Są dwa różne obliczenia parzystości – jeden XOR (jak w RAID 4 i 5), a drugi to niezależny algorytm sprawdzania danych
- Takie podejście umożliwia odtworzenie danych nawet w przypadku awarii dwóch dysków z danymi użytkownika
- Poziom RAID 6 ma tę zaletę, że zapewnia niezwykle wysoką dostępność danych. Aby dane stały się niedostępne, trzy dyski musiałyby ulec awarii w okresie równym średniemu czasowi naprawy dysku
- Poziom RAID 6 pogarsza wydajność zapisu danych, gdyż każdy zapis ma wpływ na dwa bloki parzystości