



Systemy operacyjne III

WYKŁAD 6

Jan Kazimirski



Obsługa urzędzeń wejścia i wyjścia



Urządzenia I/O

- **Komunikacja z człowiekiem** – drukarka, monitor, klawiatura, mysz.
- **Sterowane automatycznie** – dyski, napędy taśmowe, czujniki, sterowniki.
- **Urządzenia komunikacyjne** – modemy, karty sieciowe.



Klasyfikacja urządzeń I/O

- Szybkość przesyłania danych
 - wolne – klawiatura, mysz – 10^2 bajtów/sek.
 - szybkie – modem, drukarka laserowa - 10^5 - 10^6 bajtów/sek.
 - bardzo szybkie – dysk, karta sieciowa - 10^8 - 10^9 bajtów/sek.



Klasyfikacja urządzeń I/O c.d.

- Sposób dostępu do danych
 - sekwencyjny – karta sieciowa
 - bezpośredni – dysk
- Tryb pracy urządzenia
 - synchroniczny – dysk, karta graficzna
 - asynchroniczny – klawiatura, karta sieciowa



Klasyfikacja urządzeń I/O c.d.

- Tryb transmisji danych
 - urządzenia znakowe – terminal, drukarka
 - urządzenia blokowe – dysk
- Tryb użytkowania
 - możliwe współdzielenie – dysk
 - konieczna wyłączność - drukarka



Klasyfikacja urządzeń I/O c.d.

- Kierunek przekazywania danych
 - urządzenia wejściowe – klawiatura, mysz
 - urządzenia wyjściowe – drukarka, monitor
 - urządzenia wejściowe i wyjściowe – dysk, karta sieciowa, modem



Obsługa urządzeń I/O - problemy

- Różnice szybkości transferu danych
- Różna złożoność obsługi
- Różne jednostki transferu (znaki, bloki)
- Różna reprezentacja danych
- Zróżnicowana obsługa błędów



Realizacja obsługi I/O

- Programowe I/O
 - Procesor zgłasza żądanie do modułu I/O
 - Moduł realizuje operację i ustawia rejestr stanu
 - Procesor okresowo sprawdza rejestr stanu czekając na realizację operacji I/O
 - **PROBLEM: Aktywne oczekiwanie!**
 - **PROBLEM: Duże zaangażowanie procesora!**



Realizacja obsługi I/O c.d.

- Obsługa I/O sterowana przerwaniem
 - Procesor zgłasza żądanie operacji I/O i kontynuuje pracę
 - Moduł I/O realizuje operację I/O i po jej zakończeniu zgłasza przerwanie
 - Procesor uruchamia odpowiednią procedurę obsługi przerwania



Realizacja obsługi I/O c.d.

- Obsługa I/O sterowana przerwaniem c.d.
 - ZALETA: Brak aktywnego oczekiwania
 - PROBLEM: Obciążenie procesora w czasie transferu danych
 - PROBLEM: Złożona obsługa przerwania (wiele przerwania, wiele urządzeń, priorytety)



Realizacja obsługi I/O c.d.

- Bezpośredni dostęp do pamięci
 - Procesor wysyła do modułu we/wy żądanie transmisji – podając urządzenie, adres bufora w pamięci i ilość danych.
 - Moduł DMA dokonuje transferu danych pomiędzy urządzeniem a pamięcią operacyjną.
 - Moduł we/wy zgłasza przerwanie informując procesor o wykonaniu żądania.



Realizacja obsługi I/O c.d.

- Bezpośredni dostęp do pamięci c.d.
 - ZALETA: Brak aktywnego oczekiwania
 - ZALETA: Brak zaangażowania procesora w transfer (bezpośredni transfer I/O – Pamięć)
 - ZALETA: Efektywny transfer dużych bloków danych.
 - PROBLEM: Sprzętowa realizacja DMA (np. dostęp do magistrali, przepustowość magistrali).



Ewolucja funkcji I/O

- 1) Procesor steruje bezpośrednio urządzeniem
- 2) Dodano kontroler we/wy – sterowanie programowe
- 3) Wykorzystanie przerwań
- 4) Moduł we/wy wykorzystuje mechanizm DMA
- 5) Moduł we/wy staje się niezależnym podsystemem. Operacje nie angażują procesora głównego
- 6) Moduł we/wy posiada własną pamięć lokalną – może być traktowany jako niezależny wyspecjalizowany komputer realizujący operacje we/wy bez udziału procesora głównego.



Podsystem I/O - wyzwania

- Wydajność
 - Operacje we/wy są zwykle bardzo wolne (w porównaniu z dostępem do pamięci)
 - Systemy wielozadaniowe częściowo niwelują ten problem (przejęcie sterowania przez inny proces)
 - Wielozadaniowość wiąże się z podsystemem wymiany – a to jest operacja we/wy
 - Szczególnie krytyczne są operacje dyskowe

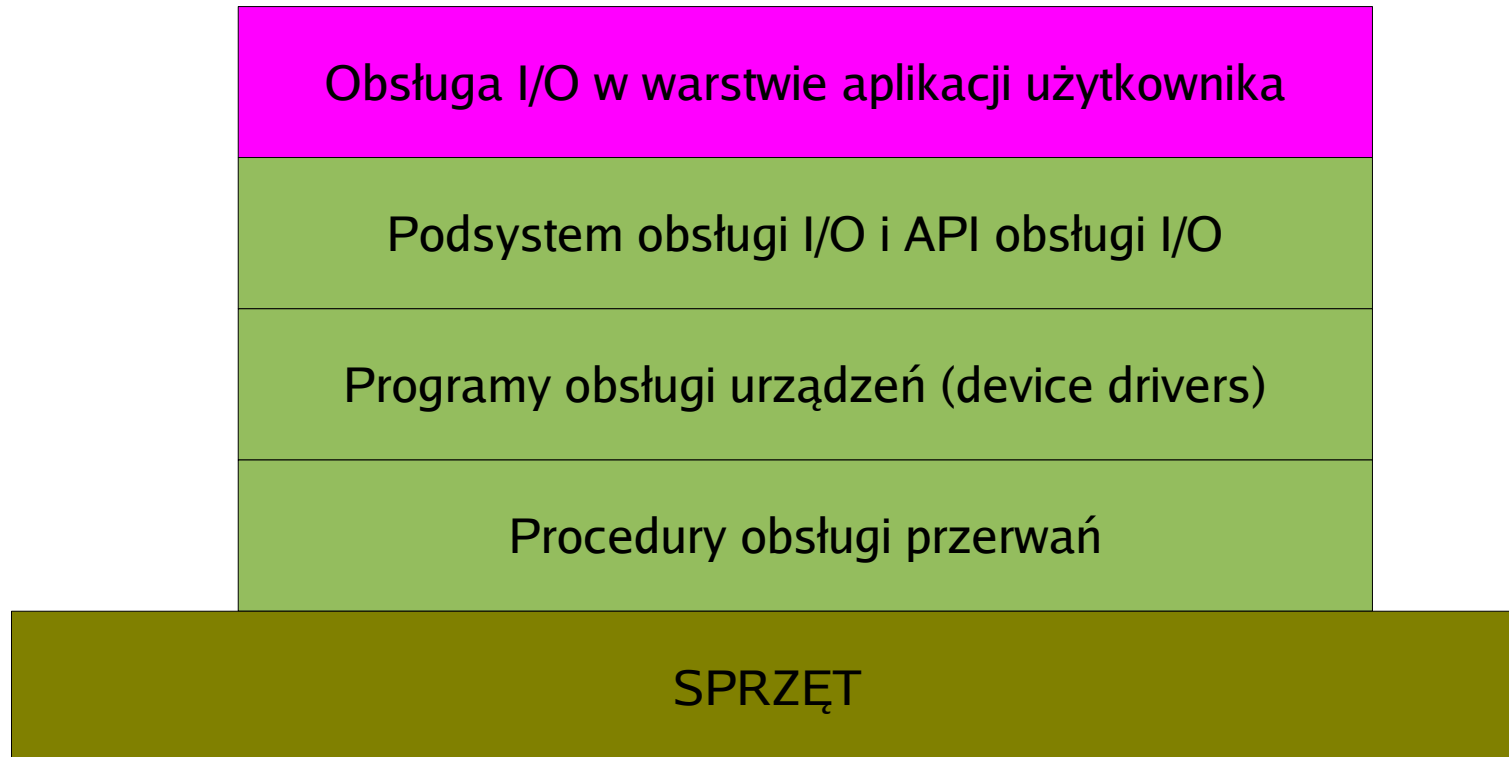


Podsystem I/O - wyzwania

- Uogólnienie
 - Spójny interfejs dostępu do wszystkich urządzeń (łatwa obsługa różnorodnych urządzeń I/O).
 - Ukrycie szczegółów poszczególnych urządzeń przed użytkownikiem (aplikacja nie musi znać szczegółów obsługi danego modelu/urządzenia I/O).
 - Stosowanie modularnego podsystemu we/wy.



Poziomy obsługi I/O





Poziomy obsługi I/O c.d.

- Procedury obsługi przerwań
 - Bardzo niski poziom abstrakcji (bezpośrednia komunikacja ze sprzętem)
 - Powinny być ukryte głęboko w strukturze SO – niedostępne dla użytkownika
 - Problemy związane ze współbieżnością i koniecznością przełączania kontekstu.



Poziomy obsługi I/O c.d.

- Programy obsługi urządzeń (sterowniki)
 - Każde urządzenie wymaga specyficznego zarządzania.
 - Program obsługi urządzenia (device driver) – kontrola specyficznego modelu urządzenia.
 - Sterowniki urządzeń dostarczane są zwykle razem z urządzeniem przez jego producenta.



Poziomy obsługi I/O c.d.

- Programy obsługi urządzeń (sterowniki) c.d.
 - Zwykle są częścią systemu operacyjnego – wbudowane lub ładowalne (moduły).
 - Stanowią dużą część kodu SO – często najbardziej kłopotliwą.
 - Projektant SO określa standard (najlepiej jednolity) komunikacji sterowników urządzeń z wyższymi warstwami SO.



Poziomy obsługi I/O c.d.

- Podsystem obsługi I/O (warstwa abstrakcji)
 - Część SO realizująca operacje I/O na wyższym poziomie abstrakcji (niezależnym od specyfiki urządzeń)
 - Typowe funkcje
 - udostępnienie jednolitego interfejsu obsługi I/O
 - buforowanie
 - obsługa błędów
 - rezerwacja urządzeń



Podsystem obsługi I/O

- Jednolity interfejs obsługi I/O
 - Jednolity zbiór operacji (read, write, ctrl)
 - Jednolite oznaczenia urządzeń
 - Jednolite zasady ochrony
- Buforowanie
 - Niwelowanie różnic szybkości urządzeń i specyfiki transferu.



Podsystem obsługi I/O c.d.

- Obsługa i raportowanie błędów
 - Dwa typy błędów: programowe i sprzętowe.
 - Jednolite zasady zachowania w przypadku wystąpienia błędu:
 - powtórzenie operacji.
 - zgłoszenie informacji o błędzie do programu. zgłaszającego żądanie.
 - wyświetlanie/logowanie komunikatów o błędach.
 - panika, zamknięcie systemu.



Podsystem obsługi I/O c.d.

- Rezerwowanie i zwalnianie urządzeń
 - Niektóre urządzenia wymagają dostępu na wyłączność.
 - Podsystem I/O rezerwuje urządzenie na czas realizacji operacji I/O
 - Żądania operacji I/O od innych programów muszą być zablokowane do czasu zwolnienia urządzenia.



Obsługa I/O w warstwie użytkownika

- Większość języków programowania zawiera funkcje obsługi I/O (jako część języka lub biblioteki).
- Programista może programować obsługę urządzeń za pomocą składni danego języka.
- Funkcje języka (lub biblioteki) odwołują się do SO w celu realizacji żądań związanych z urządzeniami I/O.



Buforowanie I/O

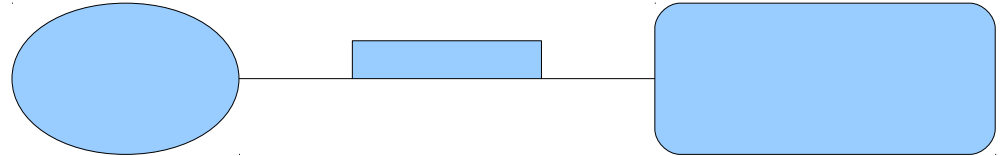
- Buforowanie = „niwelowanie różnic”
 - Dopasowanie urządzeń różniących się szybkością przekazywania danych (np. komputer → drukarka)
 - Efektywne przekazywanie bloków danych (dysk)
 - Synchronizacja współbieżnego dostępu – kopia danych dla urządzenia nie blokuje procesu

Realizacja buforowania

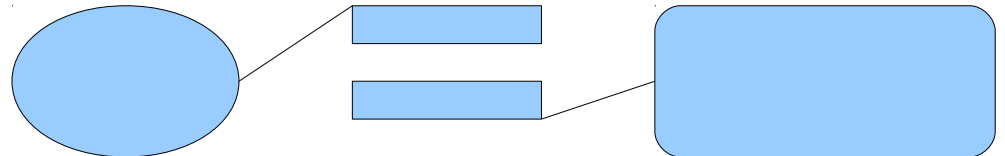
- Brak buforowania



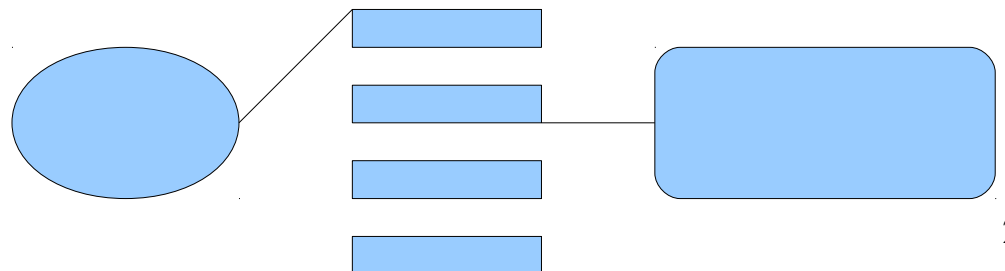
- Bufor pojedynczy



- Bufor podwójny



- Bufor cykliczny





Pamięć podręczna dysku

- Obszar pamięci przechowujący fragmenty danych z dysku.
- **Odczyt danych** – dane w pamięci podręcznej = szybki dostęp do danych
- **Zapis danych** – zwiększenie efektywności zapisów blokowych, mniejsza liczba zapisów często modyfikowanych danych



System plików

- Podstawowy element praktycznie każdego systemu operacyjnego.
- Przechowywanie danych, aplikacji (tzw. przechowywanie długoterminowe).
- Podsystem obsługi systemu plików – realizowany jako zbiór uprzywilejowanych programów lub część jądra systemu.



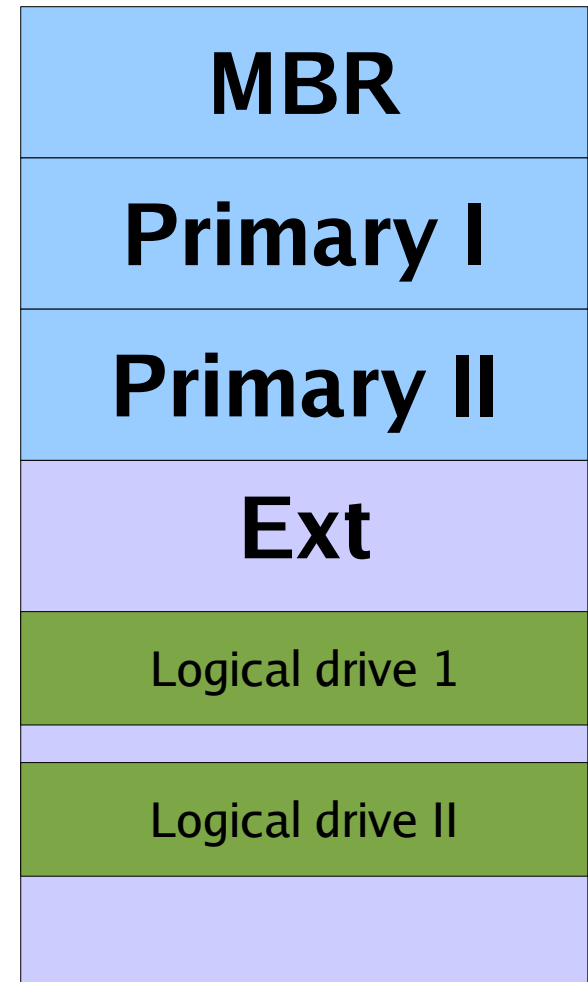
System plików c.d.

- Identyfikacja plików (nazwy).
- Umożliwia: tworzenie, usuwanie, odczyt i zapis do plików.
- Z każdym plikiem kojarzone są dodatkowe atrybuty (właściciel, uprawnienia itp.).
- Struktura hierarchiczna.



Dysk twardy komputera PC

- Sektor startowy dysku i tablica partycji (MBR)
- Partycje podstawowe (maks. 4)
- Partycja aktywna
- Partycja rozszerzona i dyski logiczne
- Boot record partycji
- System(y) plików.



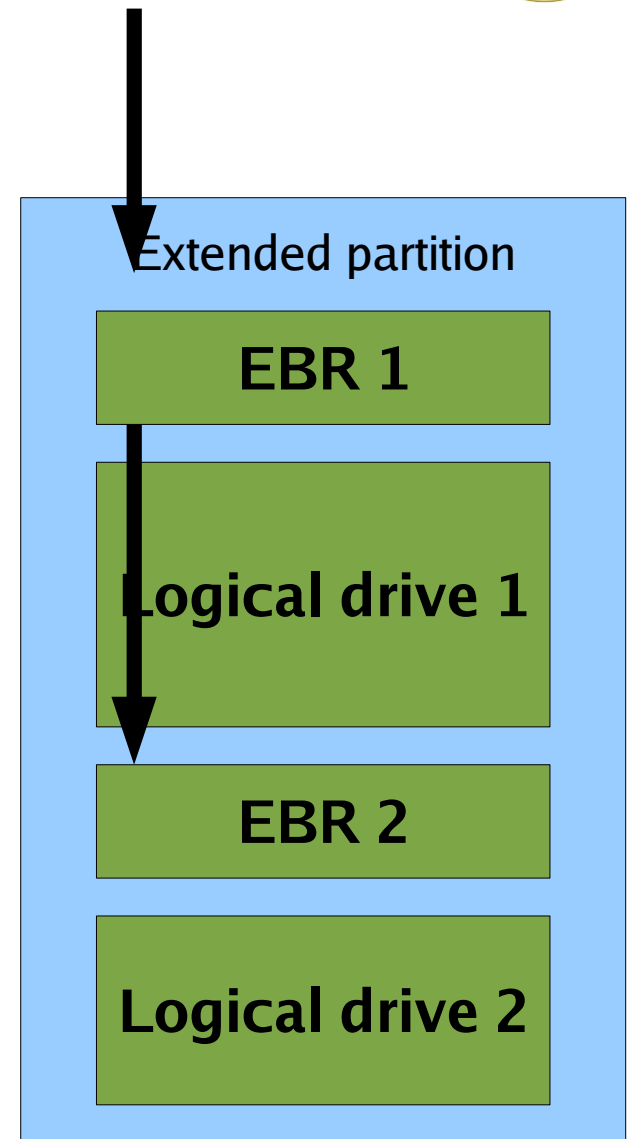


Partycje rozszerzone

- Struktura MBR-u narzuca ograniczenie na liczbę partycji na dysk (maks. 4)
- Obejściem jest “Partycja rozszerzona” zawierająca dyski logiczne.
- Strukturę partycji rozszerzonej definiują tzw. Rekordy EBR (extended boot record)

Rekord EBR

- Struktura podobna do MBR ale zawiera informację tylko o jednej partycji, oraz wskazuje na pozycję następnej partycji (chaining)
- Każdy dysk logiczny na partycji rozszerzonej zawiera swój własny rekord EBR.





Rekord startowy partycji (VBR)

- Umieszczony na początku każdej partycji
- Zawiera dane partycji oraz kod startowy (start systemu operacyjnego z danej partycji)
- Na nośnikach bez podziału na partycje (np. dyskietki) jedyny rekord startowy



Sekwencja uruchamiania systemu operacyjnego

- Po wykonaniu procedury POST BIOS ładuje do pamięci główny rekord startowy dysku (MBR).
- Sterowanie przekazywane jest do obszaru kodu tego rekordu.
- Kod z MBR analizuje tablicę partycji i wczytuje właściwy rekord startowy partycji (VBR).
- Sterowanie przekazywane jest do kodu z VBR, który ładuje jądro systemu operacyjnego.



Organizacja systemu plików

- Organizacja:
 - Płaska (CP/M)
 - Hierarchiczna (katalogi)
- Alokacja plików
 - Ciągła
 - Łańcuchowa
 - Indeksowana



Ciągła alokacja plików

- W momencie tworzenia pliku przydzielany jest ciągły obszar bloków
- Jeden wpis na plik w tablicy alokacji (blok startowy i rozmiar)
- Szybki dostęp sekwencyjny.
- Problem z tzw. “fragmentacją zewnętrzną”
- Problem przy zmianie rozmiaru pliku



Łańcuchowa alokacja plików

- Każdy blok zawiera wskaźnik do kolejnego bloku w łańcuchu
- Brak fragmentacji zewnętrznej, za to naruszona zasada lokalności (fragmentacja plików). Konieczna okresowa defragmentacja.
- Narzut danych (wskaźniki do kolejnych bloków)



Indeksowana alokacja plików

- Każdemu plikowi w tablicy alokacji odpowiada jednopoziomowy indeks bloków (przechowywany jako osobny blok).
- Brak fragmentacji zewnętrznej.
- Mniej podatna na fragmentację plików.
- Czasami łączona z alokacją łańcuchową.



Zarządzanie wolną przestrzenią

- Brak
 - Alokacja wymaga przeglądania tablic alokacji.
- Bitmapa
 - Informacja o wolnych blokach przechowywana jest w tzw. Bitmapie (1 bit na blok).
- Lista wolnych bloków lub fragmentów
 - Informacja o wolnych blokach przechowywana jest w postaci jednokierunkowej listy.



Fragmentacja

- Wewnętrzna – alokowane jest więcej miejsca niż plik potrzebuje (np. z powodu dużego rozmiaru jednostki alokacji).
- Zewnętrzna – wolny obszar na dysku jest podzielony na wiele małych obszarów w wyniku częstej alokacji i zwalniania obszarów dysku.
- Danych – dane pliku rozrzucone są w różnych oddalonych obszarach na dysku (brak lokalności danych)



Przykład systemu plików - FAT

- Stworzony w 1976 roku
- Kolejne udoskonalane wersje: FAT12, FAT16, VFAT, FAT32, exFAT
- Wykorzystywany w systemach operacyjnych DOS, Windows
- Obecnie używany w kartach pamięci.



FAT - struktura

- Boot rekord partycji i ew. sektory zarezerwowane
- Tablica alokacji plików (kopia 1)
- Tablica alokacji plików (kopia 2)
- Tablica katalogu głównego
- Obszar danych



FAT – rekord startowy partycji

- Instrukcja skoku do początku kodu startowego
- Region danych BIOS-u. Zawiera m.in.:
 - Rozmiar sektora
 - Rozmiar klastra
 - Liczba tablic alokacji plików
 - Typ nośnika
 - Dane o geometrii nośnika
 - I inne...

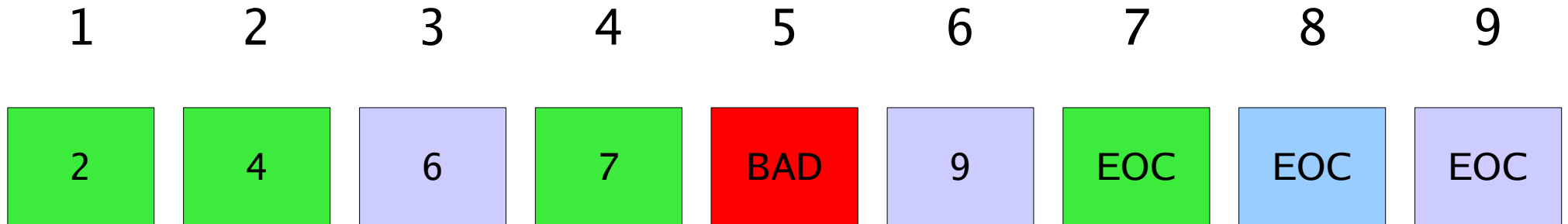


FAT – tablica alokacji plików

- Zawiera po jednym wpisie na każdy klaster dysku
- Wpis informuje czy klaster jest wolny, uszkodzony, używany.
- W przypadku używanego klastra wpis informuje o numerze kolejnego zajętego klastra danego pliku (chain) – z wyjątkiem ostatniego klastra pliku , który ma specjalne oznaczenie.



FAT – alokacja pliku



- Plik pierwszy – klastry 1,2,4,7
- Plik drugi – klastry 3,6,9
- Plik trzeci – klaster 8
- Klaster 5 – uszkodzony – zawiera sektory z uszkodzonym nośnikiem



FAT – tablica katalogu

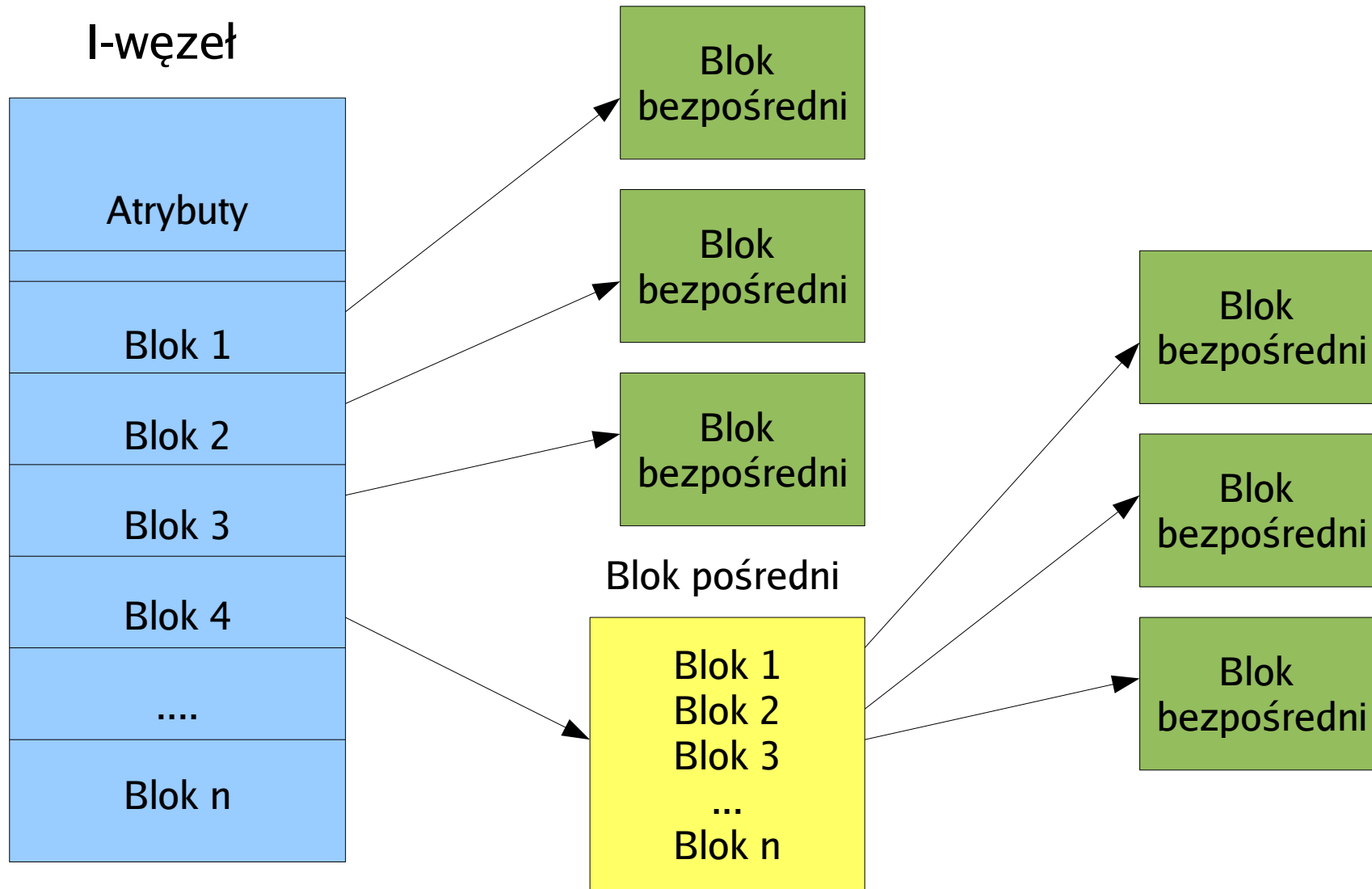
- Tablica katalogu głównego przechowywana jest w określonym regionie dysku (stały rozmiar), dla pozostałych – tablice są w obszarze danych
- Każdy wpis w tablicy przechowuje m.in.
 - Nazwę pozycji (z rozszerzeniem)
 - Atrybuty (plik/katalog, ukryty, systemowy, itd.)
 - Data i czas utworzenia, modyfikacji, dostępu
 - Numer pierwszego zajętego klastra
 - Rozmiar



Przykład Unix-owego systemu plików

- Informacje o pliku trzymane są w tzw. “i-węźle” (i-node).
- I-węzeł przechowuje listę bloków zajętych przez dany plik.
- Jeżeli plik zajmuje więcej bloków niż rozmiar listy w i-węźle to dodatkowo wprowadza się tzw. bloki pośrednie

Alokacja pliku - Unix/Linux





Rozmiar pliku

- Założenie: rozmiar bloku – 1kb, tablica bloków w i-węźle – 10 pozycji, w boku pośrednim 256 pozycji
- Możliwość alokacji(*):
 - 10 bloków bezp. 10 KB
 - Blok pośredni + 256 bloków bezp. 256 KB
 - Blok podwójnie pośredni + 256 bloków pośr. 64 MB
 - Blok potrójnie pośredni + 256 bloków podw. pośr. 16 GB

(*) Źródło: M. J. Bach *Budowa Systemu Operacyjnego UNIX*, WNT 1995