Systemy operacyjne

Operacje I/O, systemy plików

Operacje sprzętowe I/O

- systemy operacyjne zarządzają urządzeniami I/O

- jednym z wymogów przy projektowaniu systemu operacyjnego jest przechwycenie żądań I/O aplikacji i wysłanie tegoż do fizycznego urządzenia

- następnie, gdy urządzenie prześle odpowiedź system ma za zadanie przesłać ją do aplikacji

Urządzenia I/O dzielą się na dwie kategorie:

- urządzenia blokowe, które to przesyłają informacje blokowo (głównie urządzenia pamięci masowej)

- urządzenia znakowe, które przesyłają komunikaty poprzez pojedyncze znaki (bajty, oktety, słowa). Przykłady urządzeń: porty (szeregowe, równoległe), karty dźwiękowe, karty sieciowe.

Kontrolery urządzeń

- sterownik urządzenia to specjalne oprogramowanie (moduł) pozwalający systemowi operacyjnemu obsługiwać określone urządzenie

- system pośredniczy w obsłudze przerwań urządzenia

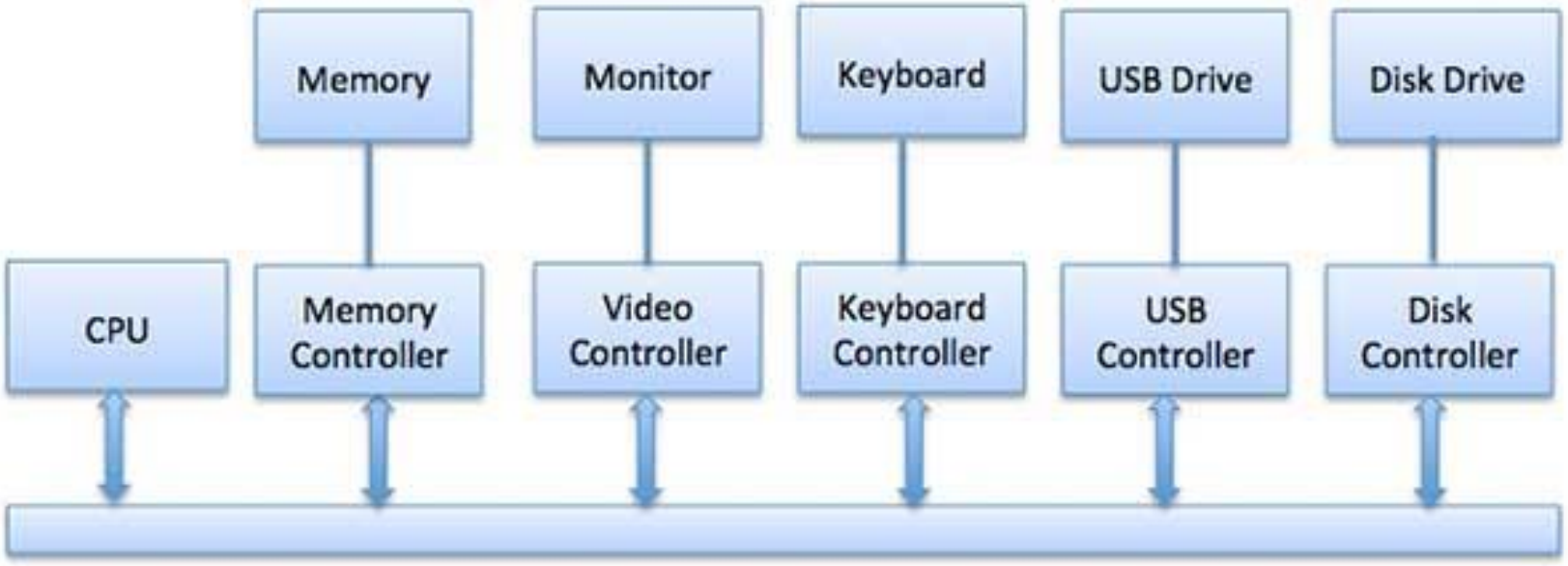
- kontroler urządzenia to pomost pomiędzy urządzeniem i sterownikiem

- niemal każde urządzenie I/O posiada określone komponenty fizyczne (przyciski, rolki, fotodiody) oraz elektroniczne komponenty obsługujące fizyczne komponenty; to właśnie elektroniczne komponenty zwane są kontrolerami

- pojedynczy kontroler urządzenia może obsłużyć wiele urządzeń. Jako interfejs konwertuje strumień bitów na bloki danych; może też korygować błędy

- urządzenia łączone są poprzez złącza i gniazda, te zaś są łączone do kontrolera.

Kontrolery urządzeń



https://www.tutorialspoint.com/operating_system/images/device_controllers.jpg

Komunikacja z urządzeniami I/O

- synchroniczne I/O - mikroprocesor czeka na przetworzenie operacji I/O

- asynchroniczne I/O - komunikacja odbywa się niezależnie od mikroprocesora

Mikroprocesor musi mieć możliwość wymiany informacji z urządzeniami I/O (niezależnie od sposobu ich działania). Na chwilę obecną istnieją trzy sposoby komunikacji:

- specjalne instrukcje I/O

- mapowanie pamięci I/O

- bezpośredni dostęp do pamięci (DMA)

Specjalne instrukcje I/O

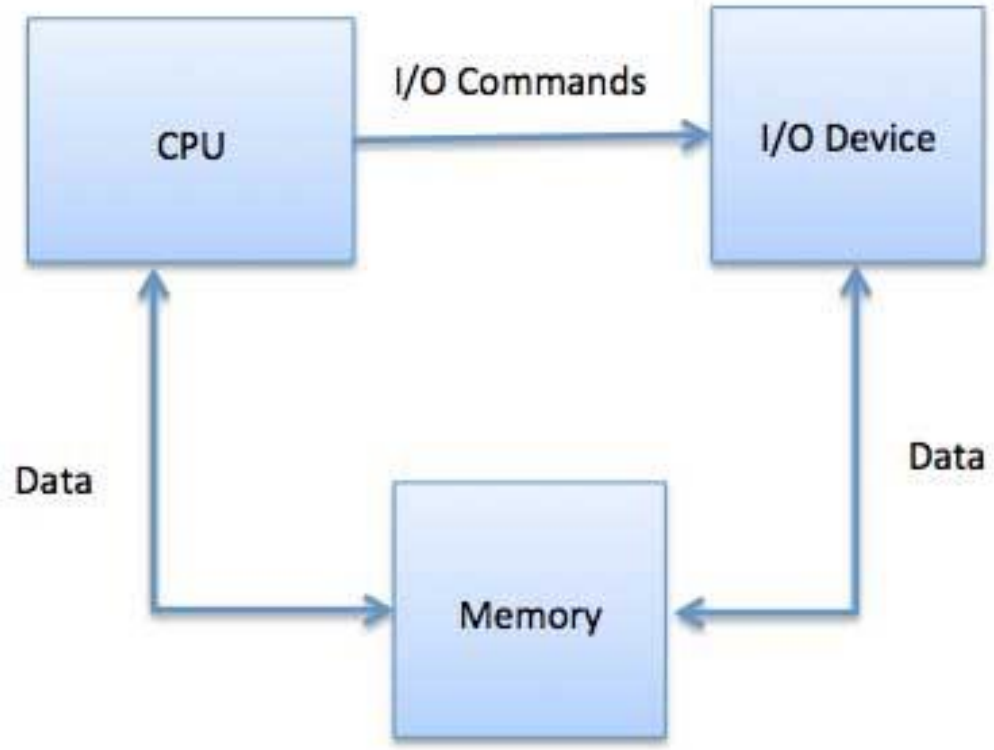
- używa instrukcji CPU stworzonych specjalnie dla kontrolowania urządzeń I/O

- instrukcje te pozwalają danym na transfer pomiędzy urządzeniem i CPU

- instrukcje to nic innego jak "rejestry", w których pojawiają się dane; te rejestry są łączone do odpowiednich fizycznych pinów CPU i urządzenia obsługiwanego

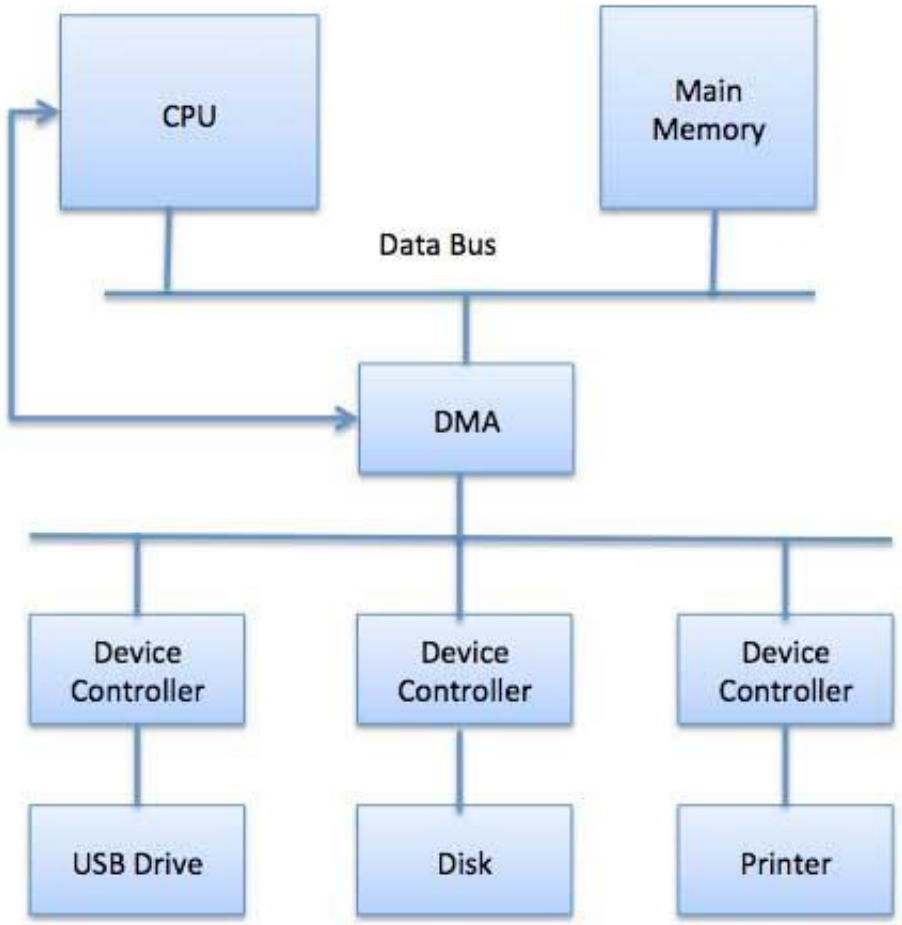
- pewnym "ucieleśnieniem" tego rodzaju instrukcji mogą być wirtualne urządzenia komputerowe

Mapowanie pamięci



https://www.tutorialspoint.com/operating_system/images/memory_mapped_io.jpg

Direct Memory Access (DMA)



https://www.tutorialspoint.com/operating_system/images/dma.jpg

Opis działania DMA

Krok	Opis
1	Sterownik dostaje polecenie przemieścić dane dysku do adresu X.
2	Sterownik komunikuje się z kontrolerem dysku do bufora pamięci.
3	Kontroler dysku uruchamia transfer DMA.
4	Kontroler wysyła bajty do kontrolera DMA.
5	Kontroler DMA transferuje bajty do bufora, zwiększając adres pamięci, zmniejszając jednocześnie licznik C.
6	Gdy C będzie posiadało wartość zero, DMA wysyła przerwanie do CPU iż transfer został zakończony.

Odpytywanie i przerwania I/O

- komputery wykrywa wejścia (dowolnego typu) na dwa sposoby - ankietowania/odpytywania oraz przerwania

- pozwala to CPU na obsługę zdarzeń zależnych jak i niezależnych od czasu czy aktualnie wykonywanych procesów

Ankietowanie/odpytywanie

- dość prosta metoda komunikacji polegająca na cyklicznym sprawdzaniu stanu urządzenia

- urządzenie podaje informacje do rejestru Statusu, zaś procesor te dane odczytuje

- urządzenia posługujące się tą techniką najczęściej nie wymagają częstej wymiany danych, obsługiwane są zaś poprzez specjalny program odpytujący

- metoda ta jest nieefektywna - marnotrawi czas procesora

Przerwania

- przerwania są sygnałami wysyłanymi od urządzeń do procesora

- CPU po otrzymaniu tego komunikatu zapisuje swój stan i przystępuje do obsługi przerwania wykorzystując tzw. wektor przerwań (specjalna przestrzeń adresów obsługi różnych zdarzeń)

- po obsłużeniu przerwania procesor wraca do poprzedniego, przerwanego zadania

Programowe I/O

- programowe wejście/wyjście ma uniezależniać ,gdzie to tylko możliwe, odwołania do urządzeń poprzez specjalnie napisane programy

- rozwiązanie to zakłada, że dzięki temu nie będzie potrzeby znać pełnej specyfikacji urządzenia - I/O zostanie obsłużone na większej ilości urządzeń bez konieczności przepisywania programu

- rozwiązanie to jest obecnie najpopularniejszym rozwiązaniem, między innymi przez zwiększenie bezpieczeństwa działania systemu operacyjnego oraz mniejszego ryzyka wystąpienia błędów/zawieszenia systemów operacyjnych.

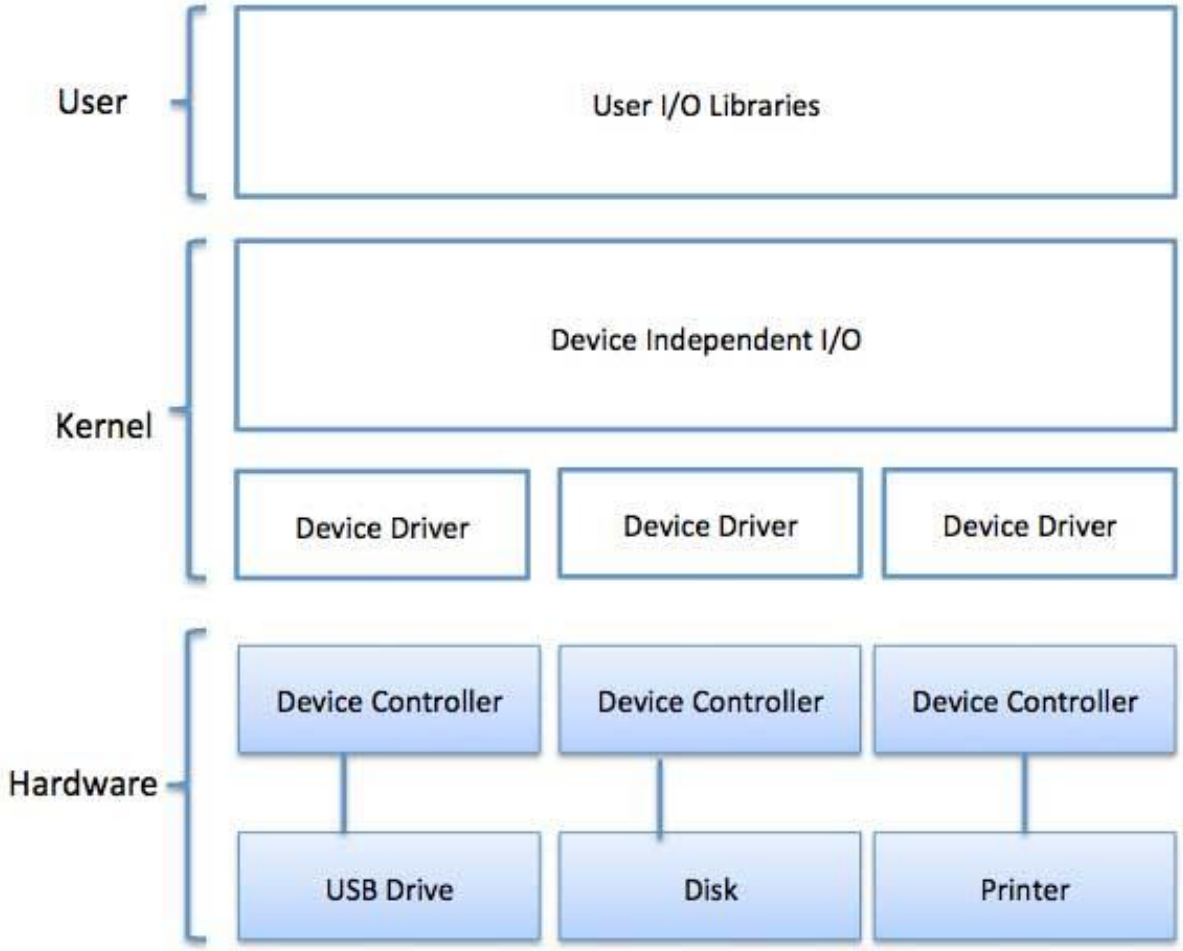
Organizacja programowego I/O

- **Biblioteki użytkownika** – dostarcza interfejs wejścia/wyjścia umożliwiający na komunikację systemu operacyjnego z użytkownikiem (niezależnie od sprzętu).

- **Moduły jądra** – dostarcza interakcję pomiędzy kontrolerem urządzenia oraz modułem I/O używanym przez sterownik.

- **Sprzęt** – warstwa gołego sprzętu, w której to kontroler wraz ze sterownikiem pozwala na realne skorzystanie z podłączonego sprzętu.

Koncepcja I/O



https://www.tutorialspoint.com/operating_system/images/io_software.jpg

Sterowniki

- to moduły podłączane do systemu operacyjnego celem obsługi określonego urządzenia

- sterownik opakowuje zależny od sprzętu kod by mógł on zostać odpowiednio odczytany przez urządzenie

- sterownik akceptuje żądania od programowego urządzenia

- komunikuje się z kontrolerem urządzenia celem nawiązania komunikacji I/O oraz obsługi błędów

Obsługa przerwań

- procedura obsługi przerwań (ISR) to oprogramowanie/funkcje wywoływane przez system operacyjny/sterownik urządzenia

- gdy nadejdzie przerwanie, procedura obsługi przerwań wykonuje w odpowiednim porządku czynności: rozpoznania przerwania, aktualizacji struktur danych, wywołania procesu czekającego na przerwanie

- mechanizm przerwań działa na adresach

- adres to numer wybierający obsługę przerwania z określonego zestawu możliwych działań. Najczęściej jest to offset (przesunięcie) przechowywane w tabeli wektorów przerwań. Wektor to adres pamięci obsługi przerwania

Programowe I/O

- obsługa jak największej ilości urządzeń poprzez te same funkcje i odwołania

- ujednolicenie odwołań już na pierwszym poziomie (użytkownika)

- najczęściej tworzone jako moduły wspólne dla szeregu określonych urządzeń (przykładowo jednakowy sterownik dla wszystkich myszy czy klawiatur)

Lista operacji programowego I/O

- jednolite interfejsy dla sterowników
- nazywanie urządzeń (mnemoniki)
- ochrona urządzenia
- niezależna wielkości bloku
- buforowanie danych (gdy dane nie mogą zostać zapisane w/na docelowym urządzeniu)
- alokacja przestrzeni przechowywania dla blokowych urządzeń
- przechwytywanie i uwalnianie określonych urządzeń
- raportowanie błędów

Programowe I/O przestrzeni użytkownika

- biblioteki tego typu pozwalają na uproszczenie dostępu do funkcjonalności na poziomie jądra lub sterownika

- zawierają procedury pozwalające na abstrakcyjną obsługę urządzeń w dowolnym programie pisanym w dowolnym języku

- przykładem tego typu rozwiązania mogą być funkcje putchar(), getchar(), printf(), scanf() ze stdio.h dla języka C

Podsystem I/O w jądrze systemu

- **harmonogram** – jądro posiada harmonogram obsługi określonych I/O. Pozwala to na ustawienie priorytetów ważności działania przykładowo z obsługą wejścia klawiatury czy myszy, a także wydruku czy wyświetlenia wyniku na monitorze.

- **buforowanie** – przestrzeń zarządzająca danymi zanim zostaną one przerzucone na właściwe urządzenie. Buforowanie pozwala niwelować problemy szybkości w czasie wymiany danych pomiędzy poszczególnymi urządzeniami.

- **przechowywanie podręczne** – fizyczne buforowanie danych na możliwie najszybszym typie pamięci.

- **kolejkowanie/rezerwacja urządzenia** – kolejka to kolejny rodzaj bufora dla urządzeń, które nie mogą przechwytywać strumienia danych. Przeważnie rozwiązanie wykorzystywane jest do wyjścia danych na konkretne urządzenie w postaci pliku. Gdy dane w pliku zostaną w całości zapisane przesyłane są jako blok danych do urządzenia.

- **obsługa błędów** – system operacyjny poprzez odpowiednią obsługę przerwań pozwala na eliminowanie występowania wielu rodzajów błędów aplikacji.

System plików - informacje wstępne

- plik (zwany zasobem) to nazwa zbioru powiązanych informacji
- informacje te to szereg bitów, mogący tworzyć dowolny zbiór (bajty, linie, słowa)
- informacje zapisywane są w dowolnej pamięci masowej

Struktura plików

- plik powinien być posiadać pewien standard zapisu danych (strukturę) zrozumiałą dla systemu operacyjnego

- na ogół pliki posiadają strukturę powiązaną do typu przechowywanej informacji; dla przykładu pliki tekstowe posiadają zapis znakowy zorganizowany w linie, pliki obiektowe (binarne) mają zaś zapis bajtów zorganizowany w bloki zrozumiałe dla sprzętu

- każdy system operacyjny wspiera typy plików zarówno wspólne dla wielu systemów jak i typy typowe dla siebie (niewspierane przez inne systemy); jeżeli system pozwala na zapis w danym formacie to przeważnie pozwala także na odczyt danego formatu

Typy plików

- typ plików wskazuje organizację tychże oraz podpowiada systemowi operacyjnemu możliwości pracy z tymi plikami

- systemy dzielą sobie rodzaje plików na następujące: pojedyncze/zwykłe, katalogi, specjalne

- pliki zwykłe przechowują informacje użytkownika, najczęściej w postaci tekstu, zbioru danych czy plików wykonywanych. Użytkownik może te pliki tworzyć, modyfikować, usuwać lub kasować.

- pliki katalogowe (foldery) zawierają listę powiązanych z nimi plików. Pliki te stanowią logiczny zbiór pojedynczych plików.

- pliki specjalne to inaczej pliki urządzeń, (możliwie) dostępnych fizycznie. Pliki te umożliwiają komunikację pomiędzy systemem operacyjnym a wspomnianymi urządzeniami. Pliki specjalne mogą mieć budowę znakową - dane obsługiwane są znak po znaku, lub budowę blokową - dane będą obsługiwane poprzez zbiór (blok) danych

Mechanizm dostępu do pliku

- mechanizm dostępu do pliku to sposób wczytywania i traktowania danych plików przez system operacyjny. Najpopularniejszymi rozwiązaniami dostępu do plików są: dostęp sekwencyjny, bezpośredni/losowy, indeksowanej sekwencji

- dostęp sekwencyjny to dostęp w pewnym porządku, jak wczytywanie linia po linii, znak po znaku. Jest to najbardziej podstawowy sposób pracy z plikami

- dostęp bezpośredni/losowy pozwala na dostęp do dowolnego fragmentu pliku. Rekordy plików posiadają własne adresy. Dzięki temu do kolejnych zapisów w pliku odwołujemy się właśnie przez adres, nie przez porządek plikowy

- dostęp indeksowanej sekwencji wymaga, by plik zawierał tabelę wskaźników do określonych bloków pliku. W tym wypadku to indeks jest przeszukiwany sekwencyjnie, zaś zawarty w indeksie wskaźnik jest używany do bezpośredniego dostępu do danych.

Przypisywanie przestrzeni (alokacja)

- przydziałem przestrzeni pamięci dla plików zajmuje się system operacyjny. Systemy posiadają trzy podstawowe możliwości przydziału przestrzeni dla plików: przyległej, połączonej, indeksowej

- przydział przyległy charakteryzuje się liniowym (przyrostowym) zajmowaniem przestrzeni pamięci poprzez przypisanie kolejnego fragmentu pamięci dla napływających danych. Najprostsza metoda, obarczona bardzo dużą fragmentacją danych

- przydział połączony tworzy listę dowiązań do zawartości danych, których zbiorczym elementem (nośną) jest plik. Foldery są w tym wypadku wskaźnikami do pierwszego bloku plików. Rozwiązanie nie sprawdza się w trybie bezpośredniego dostępu do plików

- przydział indeksowy tworzy blok indeksowy zawiera wszystkie wskaźniki do plików. Plik zawiera własny blok przechowujący adresy przestrzeni zawierające dane pliku. Foldery zawierają wskazania do bloku indeksowego plików.

Materiały i źródła

- https://www.tutorialspoint.com/operating_system

- Abraham Silberschatz, Greg Gagne, Peter B. Galvin, "Podstawy systemów operacyjnych" Tom I, Warszawa, 10 (1 w PWN), 2021