

AKADEMIA NAUK STOSOWANYCH

W NOWYM SĄCZU

Wydział Nauk Inżynieryjnych

Katedra Informatyki

OBRONA CZĘSTOCHOWY

SYSTEMY OPERACYJNE

Wielki Finał Trylogii

Autor:

Adrian "Gargi" Gargisonovsky

Prowadzący:

dr inż. Plichta Stanisława

Nowy Sącz 2025

I. Zadania i rodzaje SO.

System operacyjny to środowisko, w którym użytkownik może wykonywać programy a jego podstawowym zadaniem jest, aby był **wygodny** w użyciu i **wydajny**. Wyróżniamy **trzy** rodzaje systemów operacyjnych:

Systemy Równoległe	Systemy Rozproszone	Systemy czasu rzeczywistego
<p>-Wyposażone w wiele procesorów wykonujących obliczenia równoległe (wyróżniamy procesory symetryczne i asymetryczne),</p> <p>-Przy czym procesory mogą być:</p> <ul style="list-style-type: none">• ściśle powiązane (współdzielą magistrale, pamięć itp.),• luźno powiązane (każdy procesor posiada własną pamięć, magistrale itd.),	<p>-To szczególny przypadek systemu równoległego,</p> <p>-Wiele komputerów połączonych sieć tworzy jeden system,</p> <p>-Zalety:</p> <ul style="list-style-type: none">• Przetwarzanie bezpośrednie,• Przyspieszenie obliczeń,• Podział zasobów na prywatne i publiczne,• Przejęcie zadań uszkodzonej jednostki przez inne,• Łączność między użytkownikami,	<p>-Działa w określonych ograniczeniach czasowych,</p> <p>-Wyróżniamy dwie klasy takich systemów:</p> <ul style="list-style-type: none">• Rygorystyczne (znajduje zastosowanie jako sterownik urządzenia specjalnego przeznaczenia),• Łagodne (ma mniej napięte ograniczenia czasowe i nie zapewnia planowania w terminach nieprzekraczalnych),

Do zadań systemu operacyjnego należy:

- Zarządzanie procesami,
- Zarządzanie pamięcią operacyjną,
- Zarządzanie plikami,
- Zarządzanie systemem I/O,
- Zarządzanie pamięcią pomocniczą,
- Zapisywanie zasobów komputerowych,
- Planowanie prac,
- Ochrona zasobów,
- Umożliwienie wielodostępności,
- Umożliwienie dobrego sposobu komunikowania się z operatorem,

II. Systemy plików Windows i Linux.

System operacyjny **Windows** wykorzystuje różne systemy plików do zarządzania danymi przechowywanymi na dyskach twardych, SSD, pamięciach USB i innych nośnikach. Najważniejsze systemy plików stosowane w **Windows** to:

Windows		
FAT	NTFS	ReFS
<p>-Starszy system plików stosowany głównie w pamięciach USB i starszych systemach operacyjnych,</p> <p>-Warianty:</p> <ul style="list-style-type: none">• FAT16,• FAT32,• exFAT, <p>-FAT32 obsługuje pliki do 4 GB i partycje do 2 TB,</p>	<p>-Domyślny system plików w Windows.</p> <p>-Obsługuje duże pliki i partycje.</p> <p>-Zawiera funkcje takie jak:</p> <ul style="list-style-type: none">• uprawnienia dostępu,• szyfrowanie,• dokumentowanie zmian, <p>-Zapewnia większą stabilność i bezpieczeństwo niż FAT,</p>	<p>-Zaprojektowany do obsługi dużych systemów magazynowania danych.</p> <p>-Odporny na uszkodzenia i zoptymalizowany pod kątem wydajności.</p> <p>- Stosowany głównie w środowiskach serwerowych i macierzach dyskowych.</p>

Główne funkcje systemu plików Windows:

- Zarządzanie przestrzenią dyskową,
- Organizacja danych w katalogach i podkatalogach,
- Ochrona dostępu do plików i katalogów,
- Obsługa metadanych (np. uprawnień, daty utworzenia, atrybutów plików),
- Mechanizmy szyfrowania i kompresji danych,
- Odzyskiwanie danych po awarii systemu,
- Uprawnienia są przydzielane na poziomie użytkowników i grup z większą kontrolą dostępu,

Każdy plik jest **zbiorem danych**, które użytkownik traktuje jako pewną całość, a sam plik jest jednostką **logiczną**. System operacyjny **Linux** obsługuje wiele różnych systemów plików, dostosowanych do różnych zastosowań. Najważniejsze z nich to:

Linux		
Ext	Btrfs	XFS
<p>-Najczęściej używany system plików w systemach Linux,</p> <p>-Warianty:</p> <ul style="list-style-type: none"> • Ext2, • Ext3, • Ext4, <p>-Ext3 wprowadza dokumentowanie, co poprawia niezawodność,</p> <p>-Ext4 oferuje większą wydajność, obsługę dużych plików i lepszą optymalizację pamięci,</p>	<p>-Zapewnia zaawansowane funkcje, takie jak migawki, kompresja i kontrola integralności danych,</p> <p>-Umożliwia łatwe skalowanie systemu plików oraz zarządzanie dyskami w trybie RAID,</p>	<p>-Wysokowydajny system plików przeznaczony do dużych serwerów i dużych systemów plików,</p> <p>-Obsługuje dokumentowanie i dynamiczne alokowanie przestrzeni dyskowej,</p>

Główne funkcje systemu plików Linux:

- Obsługa wielu systemów plików w jednym systemie operacyjnym,
- Mechanizmy dokumentowania zapewniające bezpieczeństwo danych,
- Zaawansowane zarządzanie uprawnieniami użytkowników,
- Możliwość montowania systemów plików zdalnie (np. NFS, SMB),
- Wsparcie dla migawkowych kopii zapasowych i elastycznego zarządzania przestrzenią dyskową,
- Optymalizacja pod kątem wydajności i stabilności,
- Uprawnienia są przydzielane na poziomie właściciela, grupy i innych użytkowników, w postaci trzech grup znaków rwx (read, write, execute),

III. Dowiązania w systemach Windows i UNIX.

W różnych częściach systemu możemy utworzyć linki, które będą wskazywać na jeden plik. Nie musimy w ten sposób tworzyć wielu kopii tego samego pliku i możemy zaoszczędzić miejsce na dysku.

Dla Windows:

- Windows obsługuje **dowiązania symboliczne** i **dowiązania twarde** w systemie NTFS,
- **Dowiązania symboliczne** wskazują na ścieżkę **pliku** lub **katalogu**,
- **Dowiązania twarde** umożliwiają wiele nazw dla tego **samego pliku** na tej **samej partycji**,
- **Skróty** - najprostszy typ dowiązań w systemie Windows,

Dla UNIX:

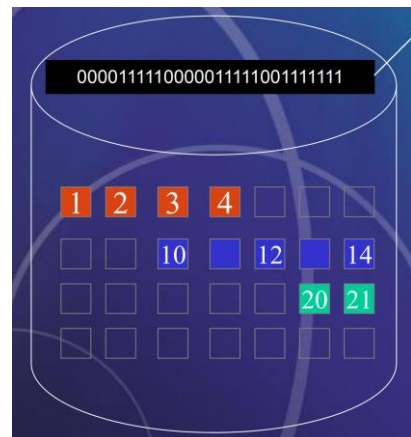
- **Dowiązania twarde** – umożliwia tworzenie kilku nazw dla jednego **i-węzła**,
- **Dowiązania symboliczne** – jest plikiem, który wskazuje na nazwę innego pliku,

IV. Sposoby zarządzania wolną przestrzenią.

Ponieważ obszar dysku jest ograniczony, więc w miarę możliwości należy dbać o wtórne zagospodarowanie dla nowych plików przestrzeni po plikach usuniętych. Lista wolnych obszarów może być implementowana w postaci:

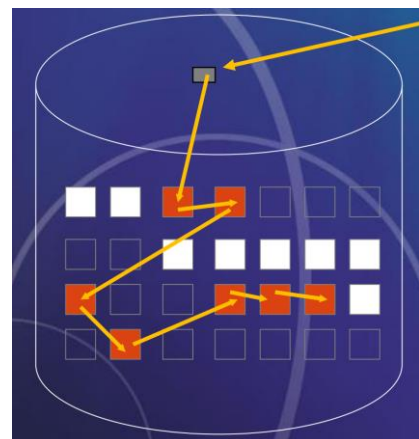
– Wektor bitowy:

- Każdy blok dyskowy jest reprezentowany przez **jeden bit** w wektorze,
- Wartość **1** oznacza, że dany blok jest **wolny**, natomiast **0** oznacza, że dany blok jest **zajęty**,
- To rozwiązanie jest **mało wydajne** i nadaje się tylko dla **małych dysków**,



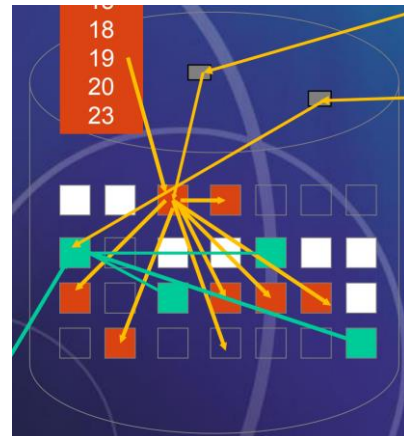
– Lista powiązana:

- Powiązanie wszystkich wolnych bloków w ten sposób, że w bloku **poprzednim** znajduje się **indeks** bloku **następnego**,
- Indeks **pierwszego** bloku znajduje się w specjalnym miejscu w systemie plików,
- To rozwiązanie jest **mało wydajne**, ponieważ aby przejrzeć listę trzeba odczytać każdy blok,



– **Grupowanie:**

- **Pierwszy wolny** blok zawiera indeksy **n** innych wolnych bloków,
- Umożliwia **szybkie** odnajdywanie większej liczby wolnych bloków,
- To rozwiązanie jest **wydajne**,



– **Zliczanie:**

- W przypadku kilku kolejnych (przylegających do siebie) wolnych bloków pamiętany jest tylko indeks **pierwszego** z nich oraz liczba wolnych bloków znajdujących się bezpośrednio **za nim**,
- To rozwiązanie jest **wydajne** dla **dużych ciągłych obszarów**,



V. Co się dzieje z procesem od jego utworzenia do zakończenia?

Stany procesu:

- **Nowy** – tworzenie procesu i przydzielenie mu zasobów,
- **Gotowy** – proces czeka na przydział procesora,
- **Aktywny** – proces otrzymał czas CPU i działa,
- **Czekający** – proces czeka na jakieś zdarzenie np. Operacje I/O,
- **Zakończony** - proces kończy działanie, zasoby są zwalniane,



Procesy przechodzą między tymi stanami zgodnie z **decyzjami planisty** i **występującymi zdarzeniami**. **System operacyjny** dynamicznie **zarządza procesami**, decydując, które mają zostać wykonane i w jakiej kolejności.

VI. Zadania planistów w systemie UNIX i Windows.

Celem planowania procesów jest jak najlepsze wykorzystanie procesora - szczególnie ważne w systemach wieloprogramowych z podziałem czasu. Planista decyduje, który proces dostanie dostęp do CPU oraz na jak długo. Planowanie odbywa się według kryteriów planowania:

– **Planowanie nie wywłaszczeniowe:**

- Proces, który dostał procesor, nie odda go aż do swego **zakończenia** lub przejścia w **stan czekania**,
- **Nie wymaga** wsparcia sprzętowego,

– **Planowanie wywłaszczeniowe:**

- **Kosztowne** – wymaga mechanizmów koordynacji,

System UNIX		
Długoterminowy	Średnioterminowy	Krótkoterminowy
<ul style="list-style-type: none">-Decyduje, które procesy mogą zostać załadowane do pamięci RAM i rozpocząć wykonywanie,-Odpowiada za liczbę aktywnych procesów,-Ważny w systemach wsadowych,-Dobranie dobrej mieszanki procesów,	<ul style="list-style-type: none">-Może wstrzymywać procesy,-Używany w systemach z ograniczoną pamięcią RAM,-Odpowiedzialny za wymianę procesów między pamięcią RAM a dyskiem,	<ul style="list-style-type: none">-Najważniejszy dla wydajności systemu,-Decyduje jaki proces zostanie wykonany przez procesor w danej chwili,-Wybiera procesy z określonym algorytmem,

W systemie Windows planowanie procesów jest realizowane przez **planistę jądra**, który zarządza przydzieleniem procesora do wątków (**Windows operuje głównie na wątkach, a nie na procesach**). Główne zadania planistów obejmują:

System Windows		
Długoterminowy	Średnioterminowy	Krótkoterminowy
<ul style="list-style-type: none"> -Decyduje, które procesy mogą zostać załadowane do pamięci operacyjnej i rozpocząć wykonywanie, -Odpowiada za liczbę aktywnych procesów, - W Windows nie istnieje wyraźnie wydzielony planista długoterminowy, ponieważ system automatycznie zarządza procesami i pamięcią w sposób dynamiczny, 	<ul style="list-style-type: none"> -Jeśli pamięć RAM się zapetnia, rzadziej używane procesy mogą być tymczasowo przenoszone na dysk, -Odpowiedzialny za wymianę procesów między pamięcią RAM a dyskiem, -W Windows planista średnioterminowy nie jest wyraźnie wydzielony, ponieważ system samodzielnie zarządza wstrzymywaniem i wznowianiem procesów w zależności od dostępnych zasobów, 	<ul style="list-style-type: none"> -Dba o efektywne przydzielanie zasobów, -Decyduje jaki proces zostanie wykonany przez procesor w danej chwili, -Priorytetowe są procesy systemowe i wątki o wysokim priorytecie użytkownika,

VII. Pojęcie wątku, czym różni się wątek od procesu?

Wątek to **najmniejsza jednostka wykonywania programu**, która działa wewnątrz **procesu**. Każdy proces może składać się z jednego lub wielu wątków współdzielących te same zasoby, takie jak pamięć czy pliki.

Proces	Wątek
<ul style="list-style-type: none">-To program w trakcie wykonywania,-Każdy proces ma własne zasoby,-Wolniejsze (trzeba przydzielić zasoby),-Procesy są od siebie izolowane,- Wymaga mechanizmów IPC (np. kolejki),	<ul style="list-style-type: none">-To jednostka wykonawcza w procesie,-Wątki współdzielą pamięć i zasoby procesu,-Szybsze (dzielą zasoby procesu),-Może odbywać się przez zmienne współdzielone,-Stosowany np. w bazach danych,

Główna różnica to fakt, że **proces to odrębna jednostka** wykonawcza z własną pamięcią, a wątek to lekka jednostka wykonawcza **działająca wewnątrz procesu**, który współdzieli jego zasoby.

VIII. Sposoby radzenia sobie z zakleszczeniami.

Zakleszczenia - to sytuacja, w której grupa procesów blokuje się nawzajem, ponieważ każdy z nich trzyma zasoby potrzebne innym i jednocześnie czeka na zasoby zajęte przez pozostałe procesy. Jest zjawiskiem niepożądanym w systemie, dlatego opracowano metody na radzenie sobie z tym. Oto główne z nich:

Zapobieganie zakleszczeniom	Unikanie zakleszczeń	Wykrywanie zakleszczeń i odtwarzanie
<p>-Zapobieganie zakleszczeniom polega na zaprzeczeniu co najmniej jednemu z czterech warunków koniecznych zakleszczenia:</p> <ul style="list-style-type: none">• Brak wzajemnego wykluczenia,• Brak przetrzymywania i oczekiwania,• Wywłaszczanie,• Wykluczenie czekania cyklicznego,	<p>-Wszystkie warunki muszą być prawdziwe,</p> <p>-Nie dopuszczamy do zakleszczeń poprzez badanie stanu systemu przed każdym żądaniem przydziału zasobów,</p> <p>-Przed każdym żądaniem sprawdza, czy jego spełnienie może doprowadzić do czekania cyklicznego,</p> <p>-Może odbywać się za pomocą np. algorytmu bankiera,</p>	<p>-System okresowo sprawdza, czy nie doszło do zakleszczenia,</p> <p>-Gdy wykryje zakleszczenie, podejmuje następujące działania:</p> <ul style="list-style-type: none">• Zabijanie procesów,• Cofanie operacji,• Stopniowe zwalnianie zasobów,

IX. Algorytmy przydziału procesora w systemach UNIX i Windows.

Algorytm przydziału procesora to metoda decydująca, który proces otrzyma dostęp do procesora oraz na jak długo. Wyróżniamy różne algorytmy w zależności od systemu. Oto kilka:

UNIX	Windows
<p>-Planowanie metodą FCFS (First Come First Served):</p> <ul style="list-style-type: none"> • Proces, który pierwszy zamówił procesor pierwszy go otrzyma, • Implementacja za pomocą kolejki FIFO, • Średni czas oczekiwania bywa bardzo długi, <p>-Planowanie metodą SJF (Shortest Job First):</p> <ul style="list-style-type: none"> • Najpierw wykonywane są procesy o najkrótszym czasie trwania, • Minimalizuje średni czas oczekiwania, • Możliwość zagłodzenia długich procesów, <p>-Planowanie metodą PS (Priority Scheduling):</p> <ul style="list-style-type: none"> • Każdy proces ma priorytet – procesor przydzielany jest procesowi o najwyższym priorytecie, • Są dwa rodzaje – z wywłaszczaniem i bez wywłaszczania, • Nisko priorytetowe procesy mogą być głodzone, <p>-Planowanie metodą RR (Round Robin):</p> <ul style="list-style-type: none"> • Każdy proces otrzymuje przydział czasu, po którym jest przesuwany na koniec kolejki, • Zapewnia sprawiedliwy podział czasu pomiędzy procesami, • Może obciążać system, 	<p>-Planowanie metodą MLFQ (Multilevel Feedback Queue):</p> <ul style="list-style-type: none"> • Procesy rozpoczynają w kolejce o wyższym priorytecie i stopniowo są przenoszone do niższych kolejek, jeśli działają długo, • Procesy interaktywne, • Sprawiedliwy podział procesora dostosowywany dynamicznie, <p>-Planowanie metodą CFS (Completely Fair Scheduler):</p> <ul style="list-style-type: none"> • Każdy proces dostaje proporcjonalny czas procesora, zależny od jego priorytetu, • Płynniejsze działanie systemu, • Nie zawsze działa optymalnie dla procesów w tle, <p>-Planowanie metodą PPS (Preemptive Priority Scheduling):</p> <ul style="list-style-type: none"> • Każdy proces ma przypisany priorytet (od 0 do 31), • Procesy o wyższym priorytecie mogą wywłaszczać procesy o niższym priorytecie, • Nisko priorytetowe procesy mogą być głodzone, <p>-Planowanie metodą RR (Round Robin):</p> <ul style="list-style-type: none"> • Każdy proces otrzymuje przydział czasu, po którym jest przesuwany na koniec kolejki, • Zapewnia sprawiedliwy podział czasu pomiędzy procesami, • Może obciążać system,

