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**
-Wyposażone w **wiele procesorów** wykonujących obliczenia równolegle (wyróżniamy procesory **symetryczne i asymetryczne**), -Przy czym procesory mogą być: • **ściśle powiązane** (współdzielą magistrale, pamięć itp.), • **luźno powiązane** (każdy procesor posiada własną pamięć, magistrale itd.),	-To szczególny przypadek systemu **równoległego**, -Wiele komputerów połączonych w sieć tworzy **jeden system**, -Zalety: • **Przetwarzanie bezpośrednie**, • **Przyśpieszenie obliczeń**, • **Podział zasobów na prywatne i publiczne**, • **Przejęcie zadań uszkodzonej jednostki przez inne**, • **Łączność między użytkownikami**,	-Działa w określonych **ograniczeniach czasowych**, -Wyróżniamy **dwie klasy** takich systemów: • **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**
-Starszy system plików stosowany głównie w **pamięciach USB** i starszych systemach operacyjnych, -Warianty: • **FAT16,** • **FAT32,** • **exFAT,** -FAT32 obsługuje pliki do **4 GB** i partycje do **2 TB,**	-Domyślny system plików w Windows. -Obsługuje duże pliki i partycje. -Zawiera funkcje takie jak: • **uprawnienia dostępu**, • **szyfrowanie,** • **dokumentowanie zmian,** -Zapewnia większą **stabilność** i **bezpieczeństwo** niż FAT,	-Zaprojektowany do obsługi **dużych systemów** magazynowania danych. **-Odporny** na uszkodzenia i zoptymalizowany pod kątem wydajności. - Stosowany głównie w **środowiskach serwerowych** i **macierzach dyskowych**.

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	**Brtfs**	**XFS**
-Najczęściej używany system plików w systemach Linux, -Warianty: • **Ext2,** • **Ext3,** • **Ext4,** **-Ext3** wprowadza dokumentowanie, co poprawia niezawodność, **-Ext4** oferuje większą wydajność, obsługę dużych plików i lepszą optymalizację pamięci,	-Zapewnia zaawansowane funkcje, takie jak migawki, kompresja i kontrola integralności danych, -Umożliwia łatwe skalowanie systemu plików oraz zarządzanie dyskami w trybie **RAID**,	-Wysokowydajny system plików przeznaczony do **dużych serwerów** i **dużych systemów plików**, -Obsługuje dokumentowanie i dynamiczne alokowanie przestrzeni dyskowej,

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:

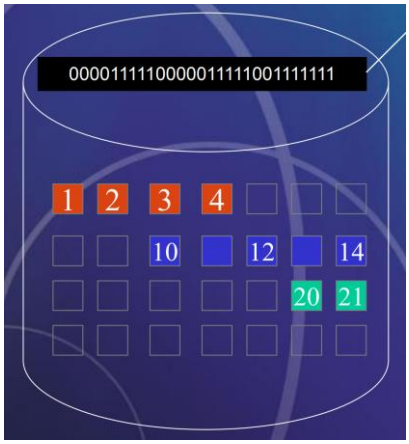
- **Dowiązania twarde** – umożliwia tworzenie kilku nazw dla jednego **i-węzła**,
- **Dowiązania symboliczne** – jest plikiem, który wskazuje na nazwę innego pliku,

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:

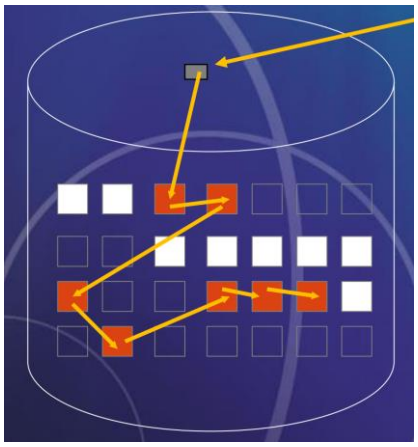
- **Wektor bitowy:**

  - 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**,



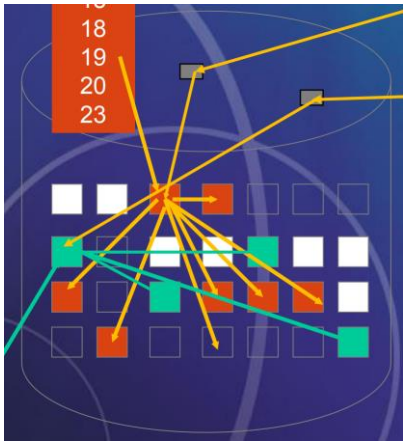
- **Lista powiązana:**

  - 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**,



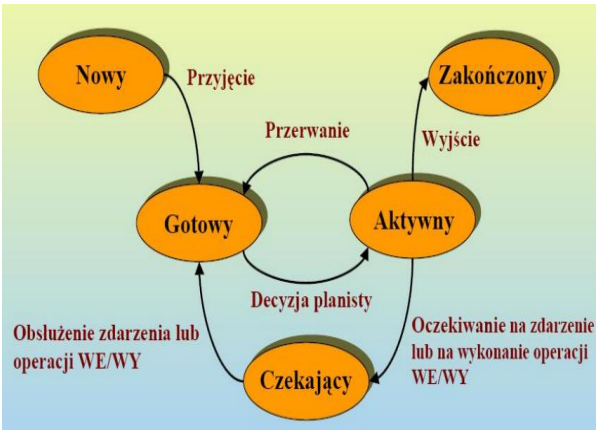
– **Zliczanie:**

- 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**
-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**,**	-Może **wstrzymywać** procesy, -Używany w systemach z ograniczoną **pamięcią RAM**, -Odpowiedzialny za **wymianę procesów** między pamięcią **RAM** a **dyskiem**,	-Dba o **efektywne** przydzielanie zasobów, -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 przydziałem procesora do wątków **(Windows operuje głównie na wątkach, a nie na procesach)**. Główne zadania planistów obejmują:

System Windows		
Planista procesów	**Planista wątków**	**Planista czasu**
- Odpowiada za przydzielanie **czasu CPU** dla **procesów,** - **Priorytety procesów** decydują o dostępie do procesora, - **Preemptive scheduling**, proces o wyższym priorytecie może przerwać aktualnie wykonywany proces, - Procesy są klasyfikowane według stanów: • **Running** – działa na CPU, • **Ready** – gotowy do wykonania, • **Waiting** – czeka na zasoby, • **Terminated** – zakończony,	- Windows używa planowania wątkowego – **wątki jednego procesu mogą działać równocześnie na wielu rdzeniach,** - Każdy proces ma co najmniej **jeden wątek,** - Planista decyduje, które wątki dostają dostęp do **CPU,** - Wątki mają priorytety i mogą być **wywłaszczane,**	- Obsługuje procesy o **najwyższym priorytecie** (np. sterowniki, multimedia), - Procesy mogą określać wymagany czas **CPU** (time constraints), - Windows stosuje **Soft Real-Time** – nie gwarantuje czasu wykonania, ale stara się go spełnić,

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
-To program w trakcie wykonywania, -Każdy proces ma **własne zasoby**, -**Wolniejsze** (trzeba przydzielić zasoby)**,** -Procesy są od siebie **izolowane,** - Wymaga mechanizmów **IPC** (np. kolejki)**,**	-To **j**ednostka 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**
-Zapobieganie zakleszczeniom polega na zaprzeczeniu co najmniej jednemu z czterech warunków koniecznych zakleszczenia: • **Brak wzajemnego wykluczenia,** • **Brak przetrzymywania i oczekiwania,** • **Wywłaszczanie,** • **Wykluczenie czekania cyklicznego,**	-Wszystkie warunki muszą być **prawdziwe**, -Nie dopuszczamy do zakleszczeń poprzez badanie stanu systemu przed każdym żądaniem przydziału zasobów, -Przed **każdym** żądaniem sprawdza, czy jego spełnienie może doprowadzić do czekania cyklicznego, -Może odbywać się za pomocą np. **algorytmu bankiera**,	-System okresowo sprawdza, czy nie doszło do zakleszczenia, -Gdy wykryje zakleszczenie, podejmuje następujące działania: • **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
-Planowanie metodą FCFS (First Come First Served): • **Proces, który pierwszy zamówił procesor pierwszy go otrzyma,** • **Implementacja za pomocą kolejki FIFO,** • **Średni czas oczekiwania bywa bardzo długi,** **-Planowanie metodą SJF** (Shortest Job First): • **Najpierw wykonywane są procesy o najkrótszym czasie trwania,** • **Minimalizuje średni czas oczekiwania,** • **Możliwość zagłodzenia długich procesów,** **-Planowanie metodą PS** (Priority Scheduling): • **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,** **-Planowanie metodą RR** (Round Robin): • **Każdy proces otrzymuje przydział czasu, po którym jest przesuwany na koniec kolejki,** • **Zapewnia sprawiedliwy podział czasu pomiędzy procesami,** • **Może obciążać system,**	**-Planowanie metodą MLFQ** (Multilevel Feedback Queue): • **Procesy rozpoczynają w kolejce o wyższym priorytecie i stopniowo są przenoszone do niższych kolejek, jeśli działają długo,** • **Procesy interaktywne,** • **Sprawiedliwy podział procesora dostosowywany dynamicznie,** **-Planowanie metodą CFS** (Completely Fair Scheduler): • **Każdy proces dostaje proporcjonalny czas procesora, zależny od jego priorytetu,** • **Płynniejsze działanie systemu,** • **Nie zawsze działa optymalnie dla procesów w tle,** **-Planowanie metodą PPS** (Preemptive Priority Scheduling): • **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,** **-Planowanie metodą RR** (Round Robin): • **Każdy proces otrzymuje przydział czasu, po którym jest przesuwany na koniec kolejki,** • **Zapewnia sprawiedliwy podział czasu pomiędzy procesami,** • **Może obciążać system,**

X. Komunikacja między procesami w systemach Windows i Unix.

Wyróżniamy różne metody komunikacji pomiędzy procesami w zależności od scenariusza, wymagań aplikacji oraz systemu operacyjnego. Oto główne z nich:

<table>
<tr><th colspan="2">Unix</th></tr>
<tr>
<td>

-Pamięć współdzielona:

- **Najszybszy** sposób komunikacji pomiędzy procesami,
- Polega na tym, że ten sam obszar pamięci jest przydzielany dla kilku procesów,

-Łącza komunikacyjne (nienazwane)**:**

- Używane do komunikacji między procesami **spokrewnionymi,**
- Jeden z procesów zamyka łącze do czytania a drugi do pisania**,**
- Pozwalają tylko na **jednokierunkowe** przesyłanie danych**,**
- **Powolny** sposób komunikacji**,**

-Łącza nazwane (kolejki **FIFO**):

- Skojarzone z nazwą ścieżki przez co mają dostęp do **niespokrewnionych** procesów**,**
- Pozwalają tylko na **jednokierunkowe** przesyłanie danych**,**
- **First in first out**, czyli pierwszy na wejściu pierwszy na wyjściu**,**

</td>
<td>

-Przesyłanie komunikatów:

- W skład narzędzi komunikacji międzyprocesowej wchodzą dwie operacje **nadaj** i **odbierz** (komunikat),
- **Uprawnione procesy** mogą pobierać komunikaty z tej kolejki,
- Komunikaty mogą mieć **zmienną** lub **stałą** długość**,**

-Sygnały:

- Procesy wysyłają sobie sygnały w celu synchronizacji (np. **SIGKILL**)

-Semafory:

- Używane do **synchronizacji** dostępu do zasobów współdzielonych**,**

-Pliki:

- Procesy mogą **wymieniać** informacje poprzez pliki zapisane w **systemie plików**,

</td>
</tr>
</table>

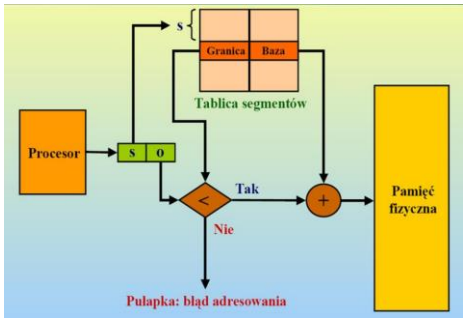
Windows	
-Pamięć współdzielona (TO SAMO CO WYŻEJ), **-Łącza komunikacyjne (TO SAMO CO WYŻEJ),** **-Łącza nazwane (TO SAMO CO WYŻEJ),** **-COM (Component Object Model):** • Pozwala na komunikację między **obiektami**, • Obsługuje komunikację **lokalną** i **zdalną**,	**-Przesyłanie komunikatów:** • Przesyła wiadomości pomiędzy wątkami i procesami za pomocą **PostMessage** i **SendMessage**, **-Semafory:** • **Kontrolują** liczbę procesów mogących uzyskać dostęp do zasobu**,**

XI. Strategie dynamicznego przydziału pamięci.

Strategie dynamicznego przydziału pamięci to metody i techniki wykorzystywane przez systemy operacyjne do efektywnego zarządzania pamięcią w czasie rzeczywistym, dostosowując alokację pamięci do potrzeb uruchomionych procesów. Oto główne strategie:

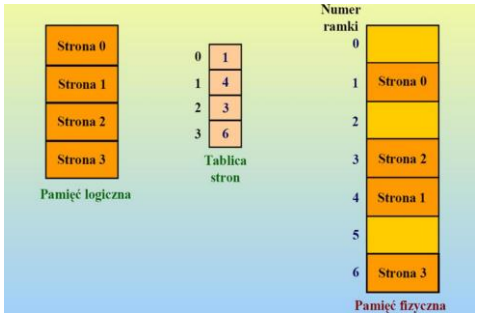
- **Segmentacja pamięci:**

  - Przestrzeń adresów **logicznych** jest zbiorem segmentów, gdzie segment to np. **funkcje**.
  - Segmenty są **ponumerowane**,
  - Odwzorowuje dwuwymiarowe adresy **logiczne** w jednowymiarowe adresy **fizyczne**,
  - Składa się z dwóch części - **baza segmentu** i **granica segmentu**,



- **Stronicowanie pamięci:**

  - Eliminuje fragmentację **zewnętrzną**, ale pozostawia fragmentację **wewnętrzną**.
  - **Adres logiczny** składa się z dwóch części:
    - **Numer strony** (indeks w tablicy),
    - **Odległość na stronie** (daje adres fizyczny pamięci),



Strategie		
First-Fit	**Best-Fit**	**Worst-Fit**
-Przydziela pamięć w **pierwszej wolnej** przestrzeni, która jest potrzebna dla procesu, -Prosta i szybka**,** bo **nie wymaga** przeszukiwania wszystkich bloków pamięci -Może prowadzić do fragmentacji **zewnętrznej**,	-Przydziela pamięć **najmniejszemu** dostępnemu blokowi, która jest potrzebna dla procesu, **-Minimalizuje** marnowanie pamięci, -Może prowadzić do fragmentacji **zewnętrznej**,	-Przydziela pamięć **największemu** dostępnemu blokowi, która jest potrzebna dla procesu, -Zmniejsza fragmentację **zewnętrzną,** -Może prowadzić do fragmentacji **wewnętrznej**,

XII. Zarządzanie pamięcią w systemie Windows i Linux.

Zarządzanie pamięcią w systemach operacyjnych Linux i Windows różni się pod względem podejścia, metod implementacji oraz mechanizmów optymalizacyjnych. Poniżej znajduję się porównanie kluczowych aspektów:

- Linux:

  - Zarządzanie pamięcią w systemie Linux odbywa się przy pomocy mechanizmów takich jak stronicowanie, system plików wymiany (swap) oraz zaawansowane algorytmy alokacji pamięci. Linux dzieli przestrzeń adresową na przestrzeń jądra oraz przestrzeń użytkownika, co zapewnia stabilność i bezpieczeństwo systemu. Pamięć operacyjna jest podzielona na strony, które mogą być dynamicznie przenoszone między pamięcią RAM a plikiem wymiany, gdy system zaczyna odczuwać jej niedobór. Administrator może kontrolować zużycie pamięci przez aplikacje, które pozwala na definiowanie limitów pamięci dla procesów.

- Windows:

  - Windows zarządza pamięcią w podobny sposób, ale działa bardziej automatycznie, żeby użytkownik nie musiał się tym przejmować. System dynamicznie przydziela pamięć aplikacjom i przechowuje często używane programy w podręcznej pamięci RAM, dzięki czemu uruchamiają się szybciej. Gdy brakuje miejsca, Windows przenosi mniej potrzebne dane do specjalnego pliku wymiany, zamiast od razu zamykać aplikacje. Dodatkowo Windows wykorzystuje kompresję pamięci, co oznacza, że może przechowywać więcej danych w pamięci RAM bez konieczności korzystania z dysku.

XIII. Algorytmy wymiany stron.

Algorytmy wymiany stron są stosowane w systemach operacyjnych do zarządzania pamięcią wirtualną, gdy brakuje wolnych stron w RAM i trzeba zwolnić miejsce poprzez przeniesienie niektórych danych do pamięci wirtualnej. Celem tych algorytmów jest wybór takich stron do usunięcia, aby minimalizować spadek wydajności systemu. Wyróżniamy kilka takich algorytmów:

- **Algorytm FIFO** (First In First Out):
  - Najprostszy sposób wymiany stron polega na usuwaniu najstarszej strony, czyli tej, która była załadowana najwcześniej. Działa jak kolejka – strona, która weszła pierwsza, jest pierwsza do usunięcia. Jest **prosty w implementacji**, ale jego wadą jest to, że może usuwać często używane strony, co prowadzi do **anomalii Belady'ego** – wzrostu liczby błędów stron mimo większej ilości dostępnej pamięci.
- **Algorytm LRU** (Least Recently Used):
  - System usuwa stronę, która była najdłużej nieużywana, zakładając, że jeśli coś nie było używane przez dłuższy czas, to prawdopodobnie nie będzie potrzebne w najbliższej przyszłości. Ten algorytm jest znacznie **skuteczniejszy niż FIFO**, ale jego **implementacja odbywa się za pomocą liczników lub stosu.** Wolny od **anomalii Belady'ego**. Jego uproszczoną wersją jest **algorytm bitów odniesienia**.
- **Algorytm LFU** (Least Frequently Used):
  - Ten algorytm usuwa stronę, która była najrzadziej używana w danym okresie czasu. Opiera się na założeniu, że strony używane rzadko są mniej potrzebne niż te często wykorzystywane. Może jednak prowadzić do sytuacji, w której niektóre strony pozostają w pamięci zbyt długo, jeśli kiedyś były intensywnie wykorzystywane, ale już nie są.
- **Algorytm OPT** (Optimal):
  - Teoretycznie **najlepszy algorytm** – usuwa stronę, która będzie potrzebna najpóźniej w przyszłości. Problem polega na tym, że wymaga znajomości przyszłych odwołań do pamięci, co w praktyce jest niemożliwe. Ma najniższy współczynnik braków stron i jest wolny od **anomalii Belady'ego**.
- **Algorytm Second Chance** (Zegarowy):
  - Strony przeglądane są w porządku **FIFO**, która daje każdej stronie „**drugą szansę**". Zamiast od razu usuwać stronę, sprawdza, czy była ostatnio używana – jeśli tak, to system daje jej kolejną szansę i przechodzi do następnej. Skuteczny i łatwy w implementacji. **Sprawdza bit odniesienia** – dla 0 (zastępuje), dla 1 (druga szansa). Strona często używana nie będzie nigdy zastąpiona.

XIV. Metody przydziału ramek.

W systemach wieloprogramowych ramki mogą być współdzielone przez kilka procesów, procesy rywalizują o te ramki. Istnieje pewna minimalna ilość ramek jaka musi zostać przydzielona procesowi. Ilość ramek musi być odpowiednia do zalokowania wszystkich stron. Schemat przydziału ramek:

- **Równy -** Każdy proces dostaje równą ilość ramek,
- **Proporcjonalny** - Proces dostaje proporcjonalną ilość ramek do jego rozmiaru,
- **Przydział priorytetowy** - Procesowi przydziela się proporcjonalną ilość ramek do jego priorytetu,

Zastępowanie globalne:

- Proces może wybierać ramki do zastąpienia ze zbioru globalnego, jeden proces może zabierać ramki drugiemu,

Zastępowanie lokalne:

- Proces może wybierać ramki do zastąpienia tylko z własnego zbioru,

XV. System wejścia/wyjścia w Linux i Windows.

- **Windows:**

System Windows zapewnia aplikacjom abstrakcję urządzeń, zarówno fizycznych, jak i programowych:

- Jednakowe zabezpieczenia i nazewnictwo,
- Asynchroniczne, pakietowe operacje wejścia-wyjścia o dużej wydajności,
- Usługi pozwalające na tworzenie sterowników,
- Dynamiczne ładowanie sterowników urządzeń i usuwanie ich z pamięci,
- Obsługa technologii Plug and Play,
- Obsługa zarządzania zasilaniem,
- Obsługa wielu możliwych do zainstalowania systemów plików,
- Menedżer operacji wejścia/wyjścia to jądro systemu operacji wejścia/wyjścia,
- Działanie systemu operacji wejścia/wyjścia bazuje na pakietach,
- Windows wykorzystuje **pakiety I/O**, co oznacza, że operacje wejścia/wyjścia są reprezentowane jako **I/O Request Packets (IRP),**

– **Linux:**

- Jednakowe mechanizmy dostępu i zabezpieczeń,
- Obsługa synchronicznych i asynchronicznych operacji wejścia-wyjścia,
- Usługi pozwalające na tworzenie sterowników,
- Dynamiczne ładowanie sterowników urządzeń i usuwanie ich z pamięci,
- Obsługa technologii Plug and Play,
- Obsługa zarządzania zasilaniem,
- Obsługa wielu możliwych do zainstalowania systemów plików,
- Menedżer operacji wejścia/wyjścia to jądro systemu operacji wejścia/wyjścia,
- Działanie systemu operacji wejścia/wyjścia bazuje na kolejkowaniu żądań,
- Linux wykorzystuje struktury żądań I/O, co oznacza, że operacje wejścia/wyjścia są reprezentowane jako struktury **request,**

XVI. Windows Serwer – architektura oraz usługi sieciowe.

Kluczowe warstwy architektury Windows Serwer:

- **Jądro (Kernel):** Obsługuje niskopoziomowe operacje systemowe takie jak zarządzanie pamięcią, procesami oraz sprzętem,
- **Warstwa usług systemowych:** Zawiera komponenty zarządzające funkcjami sieciowymi, bezpieczeństwem oraz dostępem do zasobów,
- **Warstwa aplikacji i interfejsów**: Windows Serwer oferuje narzędzia takie jak Menadżer serwera czy Powershell,

Usługi Windows Serwera:

- **DNS**: Tłumaczy nazwy domenowe na adresy IP, Strefa wyszukiwania wstecznego tłumaczy Adresy IP na nazwy domen a Strefa wyszukiwania do przodu odwrotnie,
- **DHCP**: Służy do automatycznego przydzielania adresów IP, i innych ustawień sieciowych takich jak: maski podsieci, bramy domyślne, serwery DNS,
- **Hyper-V**: Pozwala na tworzenie i zarządzanie wirtualnymi maszynami na jednym fizycznym serwerze,
- **Wirtualizacja serwerów**: Polega na uruchamianiu wielu instancji serwerów na jednym fizycznym sprzęcie co zwiększa wydajność i optymalizuje wykorzystanie zasobów,

XVII. Systemy operacyjne urządzeń mobilnych.

System operacyjny urządzeń mobilnych to specjalistyczne oprogramowanie zarządzające zasobami smartfonów, tabletów i innych urządzeń przenośnych, umożliwiające ich działanie oraz interakcję użytkownika z aplikacjami. Najpopularniejsze systemy to:

- **Android:**
  - System otwarty oparty na jądrze Linux napisany w języku Java,
  - Istnieją specjalne edycje: Android TV, Android Car, Android Wear,
- **IOS:**
  - System oparty na Mac OS X napisany w języku Objective - C,
  - Oryginalnie powstał dla Iphone, później rozszerzony dla iPoda, iPada, AppleTV,
- **Windows Phone:**
  - Mało popularny i niedoceniany system, napisany w języku C#,
  - Jego następcą był Windows 10 Mobile,
- **Fuchsia:**
  - System operacyjny rozwijany przez Google, oparty na jarze o nazwie “Zircon”,
  - Napisany w języku Flutter, działa na tabletach, telefonach oraz komputerach,
- **Sailfish OS:**
  - System oparty na jądrze Linux napisany w języku C++,
  - Stawia na prywatność i bezpieczeństwo użytkownika, brak szpiegowskich usług Google,
- **Tizen:**
  - System oparty na jądrze Linux napisany w języku C++ oraz C,
  - Rozwijany przez firmę SAMSUNG,