

# **Zagadnienia do egzaminu oraz odpowiedzi ( WYKŁAD)**

Scalone wykłady: [PRAWYM NA TO I OTWÓRZ LINK](#)

|  |   |
|--|---|
| 1.1. Zagadnienia – 40.....   | 1 |
| Przejawy braku integralności systemu plików:.....                              | 8 |
| - ..... Brak bloku zarówno w wykazie bloków zaalokowanych jak i bloków wolnych |   |
| .....  | 8 |

## **1.1. Zagadnienia – 40.**

### **1. Struktury systemów komputerowych**

System operacyjny, to złożony system informatyczny. Musi on mieć odpowiednią strukturę, tak aby jego zadania mogły być podzielone na mniejsze moduły o dobrze określonej funkcjonalności i interfejsach. Trzy popularne struktury to:

#### **- Struktura jednolita -**

Najdawniejsze systemy operacyjne właściwie nie mają żadnej struktury. Taki system operacyjny jest jedynie ogromnym programem zawierającym wiele procedur, które bez żadnych ograniczeń mogą się wywoływać i być wywoływane przez inne procedury. Istnieje wiele komercyjnych systemów nie mających dobrze określonej struktury. Systemy te powstawały najczęściej jako małe i proste, a dopiero później się rozrastały. Przykładem takiego systemu jest MS-DOS.

#### **- Struktura warstwowa**

Tworzenie systemu polega na podzieleniu go na moduły, które są połączone w warstwy. Każda warstwa spełnia funkcje, które zależą tylko od warstwy znajdującej się pod spodem. Podział na moduły zmniejsza stopień wzajemnych zależności między różnymi składowymi systemu. Pozwala uniknąć niepożądanych powiązań.

#### **- Struktura klient-serwer**

Podział systemu operacyjnego na moduły.

- Moduły nie są rozmieszczone w warstwach.
- Nie komunikują się między sobą poprzez wywoływanie procedur, ale wysyłają komunikaty za pośrednictwem centralnego programu obsługi komunikatów.
- Komunikaty mogą być wysyłane w obie strony.
- Moduł wysyłający początkowy komunikat nazywa się klientem.
- Moduł odbierający ten komunikat serwerem.

Przykładem systemu o takiej strukturze jest system Windows

### **2. Zadania systemu operacyjnego.**

- Zadaniem systemu operacyjnego jest tworzenie środowiska, w którym użytkownik może wykonywać programy.
- Zarządzanie efektywnością komputera.
- Zarządzanie oraz zbieranie danych umieszczonych na dyskach.
- Udostępnianie aplikacji (jako wirtualne maszyny) ...
- Jednoczesne działanie wielu aplikacji (Wielozadaniowość) ...
- Kontakt z użytkownikiem. ...
- Łączność z innymi urządzeniami bądź komputerami.

Podstawowym celem systemu operacyjnego jest spowodowanie by:

- system komputerowy był wygodny w użyciu
- eksploatacja sprzętu komputerowego była wydajna

### **3. Właściwości systemu operacyjnego.**

## **Wersja dla pojebanych (#jebaćraszple):**

System operacyjny jest istotną częścią wszystkich systemów komputerowych. Istnieje wiele definicji systemów operacyjnych, tak jak istnieje wiele systemów operacyjnych. Każda z definicji charakteryzuje system operacyjny z innej perspektywy. System operacyjny:

- Pośredniczy między użytkownikiem a sprzętem komputerowym.
- Chuj wie czy to o to chodzi Użytkownicy korzystają z systemu komputerowego poprzez rozmaite programy użytkowe. Programy te nie mają jednak bezpośredniego dostępu do sprzętu komputerowego. Korzystają tu z pośrednictwa systemu operacyjnego. System operacyjny określa sposoby korzystania z zasobów sprzętowych.
- Udostępnia programom *maszynę wirtualną*.
- System operacyjny pośrednicząc między programami i sprzętem komputerowym ukrywa wiele szczegółów technicznych przed programami. Programy widzą urządzenia w uproszczony, jednolity sposób, niezależny od konkretnego rodzaju sprzętu i niskopoziomowych szczegółów jego obsługi. W ten sposób, programy nie widzą konkretnego sprzętu, a jedynie uproszczoną i ujednoliczoną maszynę wirtualną.
- Przydziela sprzętowe zasoby systemu komputerowego programom.
- Zasoby te, to np. pamięć, procesor, urządzenia wejścia/wyjścia. Programy uruchamiane przez użytkowników potrzebują tych zasobów do wykonania realizowanych przez nie zadań. To jednak system operacyjny przydziela te zasoby. Można go przymierzać do kierownika przedsiębiorstwa. Decyduje o tym, ile i jakich zasobów ma być przydzielonych poszczególnym pracownikom tak, aby ci mogli realizować powierzone im zadania. Sam jednak nie wytwarza nic, a jedynie dba o sprawne funkcjonowanie przedsiębiorstwa. W sytuacji, gdy dochodzi do konfliktów między programami o przydzielanie zasobów, system

operacyjny musi rozstrzygać te konflikty dbając o zaspokojenie potrzeb wszystkich programów.

- Nadzoruje pracę programów.
- W sytuacji, gdy w systemie komputerowym działa wiele programów, nieprawidłowe działanie programu może zakłócić działanie innych programów lub wręcz całego systemu komputerowego. System operacyjny przeciwdziała temu. Pilnuje, aby programy korzystały prawidłowo z powierzonych im zasobów, aby nie ingerowały w zasoby przydzielone innym programom i przeciwdziała ewentualnym błędom. System operacyjny odgrywa tu rolę policjanta.
- Tworzy wygodne i bezpieczne środowisko pracy dla użytkowników.
- System operacyjny (wraz z programami użytkowymi) tworzy środowisko pracy użytkownika. W dużej mierze, to od systemu operacyjnego zależy, czy użytkownik będzie czuł się wygodnie w tym środowisku. O tej wygodzie decyduje przede wszystkim *interfejs* użytkownika, zestaw podstawowych narzędzi udostępnianych przez system, wydajność systemu komputerowego, jego niezawodność i bezpieczeństwo.
- Przechowuje dla użytkowników rozmaite informacje.
- Zadania wykonywane przez systemy komputerowe wiążą się (przeważnie) z przetwarzaniem informacji. Użytkownicy chcą przechowywać w systemie komputerowym rozmaite informacje. Do systemu operacyjnego należy ułatwienie im tego, stworzenie odpowiedniego mechanizmu magazynowania informacji (np. system plików) wraz z podstawowymi narzędziami.

## **Wersja dla ubogich:**

### **System operacyjny:**

- Pośredniczy między użytkownikiem a sprzętem komputerowym.
- określa sposoby korzystania z zasobów sprzętowych.
- Udostępnia programom *maszynę wirtualną*.
- ukrywa szczegóły techniczne przed programami. Programy widzą urządzenia w uproszczony, jednolity sposób.
- Przydziela sprzętowe zasoby systemu komputerowego programom.
- Decyduje o tym, ile i jakich zasobów należy przydzielić programom, aby mogły dobrze realizować powierzone im zadania.
- Nadzoruje pracę programów.
- System operacyjny pilnuje, aby programy korzystały prawidłowo z powierzonych im zasobów, aby nie ingerowały w zasoby przydzielone innym programom i przeciwdziała ewentualnym błędom.
- Tworzy wygodne i bezpieczne środowisko pracy dla użytkowników.

- Przechowuje dla użytkowników rozmaite informacje.

#### **4. Pożądane cechy systemu operacyjnego.**

- funkcjonalność
- wydajność,
- niezawodność
- bezpieczeństwo
- Rekonfigurowalność - tak wiem pojęte słowo - przecież Staszka nawet nie wie co ono oznacza

#### **5. Zadania jądra systemu.**

Jądro składa się z zestawu funkcji zawartych w bibliotece Ntoskrnl.exe, które realizują podstawowe mechanizmy systemu:

- szeregowanie wątków,
- usługi synchronizacyjne wykorzystywane przez składniki centrum wykonawczego,
- niskopoziomową obsługę przerwań i wyjątków.

Jądro jest warstwą łączącą sprzęt z oprogramowaniem. Realizuje najczęściej wykonywane zadania w systemie operacyjnym, np. wykonanie zapisu pliku, odczyt danych z pamięci, przetwarzanie danych otrzymanych z karty sieciowej itp.

Przykładowe zadania jądra systemu operacyjnego:

- obsługa procesów (uruchamianie, wstrzymywanie, przełączanie, zatrzymywanie, ...) — najważniejsze zadanie jądra,
- obsługa urządzeń i ich przerwań,
- zarządzanie zasobami komputera,
- zarządzanie pamięcią,
- zarządzanie pamięcią masową,
- obsługa sieci.

#### **6. Udogodnienia sprzętowe.**

Aby programy użytkownika nie mogły dezorganizować pracy systemu, sprzęt musi mieć odpowiednie mechanizmy gwarantujące właściwe zachowanie się całości, czyli tzw. udogodnienia sprzętowe:

- Mechanizm przerwań
- Ochrona pamięci operacyjnej
- Zbiór rozkazów uprzywilejowanych
- Zegar czasu rzeczywistego

## **7. Budowa systemu operacyjnego.**

**Każdy system operacyjny zbudowany jest z trzech, podstawowych części: jądra, systemu plików oraz powłoki.**

Każda część charakteryzuje się czym innym oraz pełni inne funkcje. Dzięki temu system operacyjny spełnia tak zróżnicowane funkcje, pozwalające na efektywne zarządzanie systemem komputerowym.

## **8. Organizacja pamięci pomocniczej.**

- Pamięć pomocnicza jest zarządzana przez system plików.
- System operacyjny dostarcza jednolitego logicznie obrazu przechowywania informacji w oderwaniu od cech fizycznych urządzeń.
- PLIK - logiczna jednostka informacji.
- System plikowy:
  - zbiór plików
  - struktura katalogów

## **Suka podpierdala od innych z Internetu.**

### **ORGANIZACJA PAMIĘCI POMOCNICZEJ**

Pamięć pomocnicza podzielona jest na bloki. Występują następujące techniki dynamicznego przydzielania bloków pamięci pomocniczej dla plików:

- ✖ system plików zwartych (przydział ciągły)
- ✖ łańcuch powiązanych bloków (przydział listowy)
- ✖ mapa plików (tablica przydziału plików)
- ✖ bloki indeksów (przydział indeksowy)

### **Metody przydziału miejsca na dysku**

- system plików zwartych (przydział ciągły)
- łańcuch powiązanych bloków (przydział listowy)
- mapa plików (tablica alokacji)
- bloki indeksów

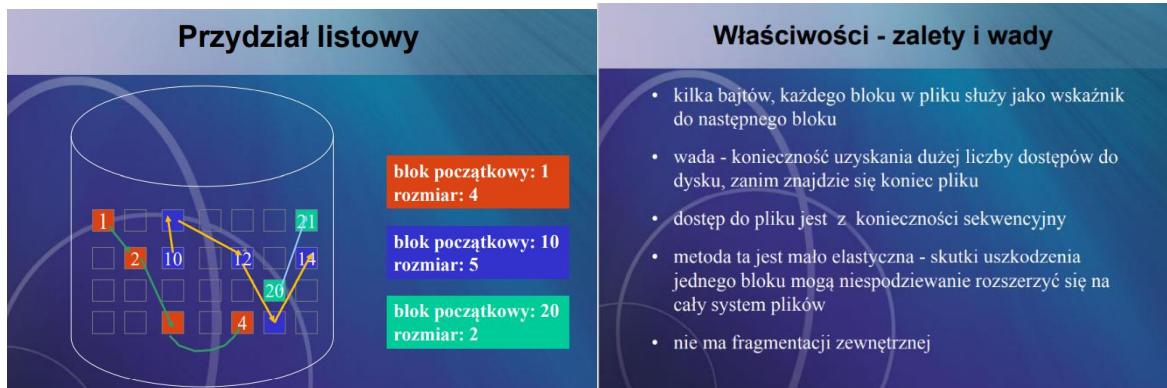
### **PRZYDZIAŁ CIĄGŁY:**



### **Właściwości - zalety i wady**

- łatwo implementować dostęp swobodny i sekwencyjny,
- trudno uniknąć fragmentacji zewnętrznej,
- umożliwia najbardziej elastyczną organizację danych - zniszczenie jednego bloku powoduje tylko lokalną utratę danych,
- odpowiednie do takich zastosowań jak bazy danych.

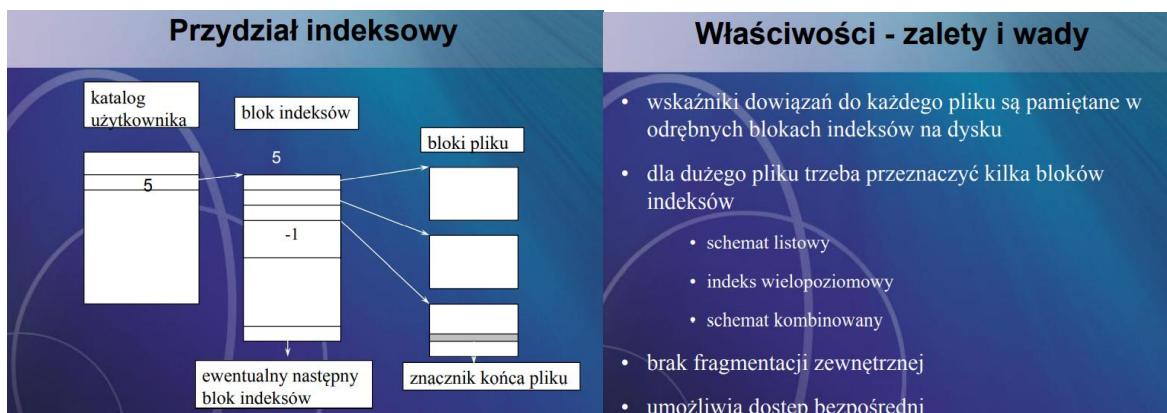
## PRZYDZIAŁ LISTOWY:



## MAPA PLIKÓW:



## PRZYDZIAŁ INDEXOWY:



## 9. Określenie rozmiaru bloku dyskowego.

## **screen JEST PRADOPODOBNE ŹLE**

- Pamięć masowa jest podzielona na 512-bajtowe sektory (2048 bajtów w CD-ROMie). Blok dyskowy jest wielokrotnością sektora. Typowe rozmiary bloku to 1,2,4,8 sektorów.

- Większy rozmiar bloku => większa prędkość transmisji (mniejsza fragmentacja, czyli mniej ruchów głowicą).
- Większy rozmiar bloku => mniej efektywne wykorzystanie miejsca na dysku. Przeciętnie z każdym plikiem wiąże się utrata miejsca równa połowie długości bloku



#### 10. Zadania systemu plikowego.

- Pozwala tworzyć i usuwać pliki.
- Umożliwia dostęp do plików w celu czytania i pisania.
- Zarządza automatycznie przestrzenią pamięci pomocniczej.
- Umożliwia odwoływanie się do plików za pomocą nazw symbolicznych.
- Chroni pliki przed skutkami uszkodzenia systemu.
- Pozwala by współpracujący użytkownicy mogli korzystać z tych samych plików, ale też chronić pliki przed dostępem do nich nieuprawnionych użytkowników

#### 11. Integralność systemu plikowego.

- W wyniku awarii systemu zawartość podręcznej pamięci buforowej może nie zostać zapisana na dysku lub może zostać zapisana tylko częściowo.
- Skutkiem w/w awarii może być pozostawienie systemu plików w stanie niespójnym.
- Większość systemów operacyjnych dostarcza odpowiednie narzędzie do sprawdzania integralności systemu plików, uruchamiane w ramach restartu systemu po awarii.

Przejawy braku integralności systemu plików:

- Brak bloku zarówno w wykazie bloków zaalokowanych jak i bloków wolnych
- Obecność bloku zarówno w wykazie bloków zaalokowanych jak i bloków wolnych
- Wielokrotne powtórzenie się bloku w wykazie bloków wolnych (duplikacja wolnego bloku)
- Wielokrotne powtórzenie się bloku w wykazie bloków zaalokowanych (duplikacja zaalokowanego bloku)
- Niespójność informacji we wpisach katalogowych (np. niezgodność licznika dowiązań w systemie UNIX).

## 12. Systemy plików.

- Każdy plik jest zbiorem danych, które użytkownik traktuje jako pewną całość.
- Plik jest jednostką logiczną, na której system plików wykonuje pewne operacje. **Operacje, które można wykonać na plikach**
  - Tworzenie pliku
  - Zapisywanie pliku
  - Czytanie pliku
  - Zmiana pozycji w pliku
  - Usuwanie pliku
  - Skracanie pliku

**Popularne systemy plików:**

**WINDOWS:**

### FAT32 -

FAT jest jednym z najstarszych spośród obecnie jeszcze używanych systemów plików. Pierwsza wersja (FAT12) powstała w 1980 roku. Wraz z wzrostem rozmiaru dysków i nowymi wymaganiami użytkowników powstawały kolejne wersje tego systemu (FAT16 w 1983, FAT32 w 1996), ale idea pozostała ciągle ta sama.

#### Podstawowe cechy

Poniższa tabela przedstawia podstawowe cechy systemów plików z rodziny FAT.

| Cecha                                     | Wartość dla                                    |                |  |
|---|--|----------------|--|
|   | FAT12  | FAT16          | FAT32 / VFAT   |
| Maksymalna długość nazwy pliku            | 8+3 bajtów                                     | 255 znaków (!) |  |
| Znaki, które mogą wystąpić w nazwie pliku | Teoretycznie wszystkie z wyjątkiem 0x00 i 0xE5 |                |  |
| Maksymalna długość ścieżki                | Brak   |                |  |
| Maksymalna długość pliku                  | 32MB   | 2GB            | 4GB  |
| Maksymalny rozmiar partycji               | 2MB - 128MB                                    | 16MB - 2GB     | 512MB - 2TB (w praktyce ok. 120GB ze względu na rozmiar FAT) |
| Rozmiar bloku                             | od 512 bajtów do 32KB                          |                |  |
| Maksymalna ilość plików                   | 4077   | 65517          | 268.435.437  |

### NTFS -

- Transakcyjny model wykonywania operacji.
- Elegancki sposób przechowywania metadanych.
- B-drzewa, bezpieczeństwo i stabilność, indeksowanie ogólne (general indexing).
- Wydane wersje NTFS:
  - NTFS v1.2 (4.0) na Windows NT
  - NTFS v3.0 (5.0) na Windows 2000
  - NTFS v4.0 (5.1) na Windows XP

Ograniczenia nałożone na pliki umieszczone w NTFS:

| Maksymalna długość nazwy pliku<br>255 bajtów | Dozwolone znaki w nazwie pliku<br>Każdy znak Unicode poza NUL | Maksymalna długość ścieżki<br>bez limitu | Maksymalny rozmiar pliku<br>16TB | Maksymalny rozmiar woluminu<br>256TB |
|--|---|--|----------------------------------|--------------------------------------|
|--|---|--|----------------------------------|--------------------------------------|

Każda operacja na systemie plików NTFS jest realizowana w postaci transakcji. Najpierw wszelkie zmiany w metadanych zostają zapisane do dziennika, potem dopiero nanoszone na dysk. Ułatwia to naprawianie błędów, odzyskiwanie systemu plików.

W NTFS, wszystkie informacje opisujące plik(nazwa pliku, data utworzenia, prawa dostępu, ...) są przechowywane jako metadane. Ten abstrakcyjny, obiektowy model pozwala na łatwe rozszerzanie funkcjonalności NTFS.

Patrząc od strony implementacyjnej, B-drzewa użyte do przechowywania systemowych danych plików(folderów) pozwalają na szybki dostęp i redukcję defragmentacji na dysku. Dziennik systemu plików zapewnia jego integralność. Wpływą też znacząco na sprawdzoną stabilność NTFS, wyjątkowo ważną przy wymienianiu głównych wad poprzednich wersji systemów plików pod np. Windows NT.

Niektoří z funkcji NTFS opierają się na wyjątkowo ważnej jego ceście: ogólnym indeksowaniem po atrybutach. Pozwala ono na sortowanie plików(folderów) po atrybutach, a następnie na szybkie wyszukiwanie. Standardowo, zawartość katalogu jest przechowywana alfabetycznie w strukturze B+ drzewa(B-drzewa z połączonymi pomiędzy liśćmi.

## 2. Struktura woluminu NTFS



### LINUX:

#### Ext / ext3 / ext4 -

### ext / ext2 / ext3 / ext4

**Ext2** (ang. *Second Extended File System*) jest podstawowym i najszerzej używanym systemem plików dla Linuxa. Jest bardzo efektywny w typowych zastosowaniach, a równocześnie stosunkowo prosty. Jest uważany za jeden z najlepszych "standardowych" systemów plików.

- Zapewnia wszystkie elementy systemu plików Unix (dowiązania symboliczne, pliki specjalne, prawa dostępu...)
- Wysoka wydajność dzięki przeciwdziałaniu fragmentacji (poprzez przydzielanie bliskich bloków oraz prealokację)
- Wydajny mechanizm dowiązań symbolicznych
- Stabilny i dobrze przetestowany
- Dobrze zdefiniowany sposób dodawania rozszerzeń
- Niezależny od tworzącego systemu operacyjnego (wszystkie pola wielobajtowe zapisane w standardzie little-endian)
- Maksymalny rozmiar partycji to 4TB, a pojedynczego pliku 2GB. Maksymalna długość nazwy pliku: 255 znaków.
- Obsługa "dziurawych" plików (nieużywane bloki nie zostają przydzielone).

### 13. Reprezentacja procesów w jądrze systemy LINUX, WINDOWS.

#### **Nie wiem czy to jest dobrze napisane :(**

Proces (nazywany też czasem zadaniem) to wykonujący się program. Wykonanie instrukcji jednego procesu musi być sekwencyjne, inaczej mówiąc instrukcje są wykonywane w określonej kolejności. Każdy proces ma własny licznik instrukcji, który wskazuje następną instrukcję do wykonania, oraz własny obszar przydzielonej mu pamięci operacyjnej. Pamięć operacyjna zwykle jest podzielona na cztery części (nazywane segmentami):

- segment kodu -- zawiera instrukcje wykonywanego programu,

- segment danych -- zawiera zmienne globalne programu,
- stos -- jest używany do wywoływania procedur, przekazywania parametrów i wyników, oraz przechowywania zmiennych lokalnych,
- sterata -- z tego obszaru przydzielana jest pamięć dla zmiennych dynamicznych.

- **Tworzenie procesu:**

- Linux: fork
- Windows: CreateProcess

- **Usuwanie procesu:**

- Linux: exit, abort, kill
- Windows: ExitProcess, TerminateProcess

- **Zmiana priorytetu procesu:**

- Linux: nice (setpriority)
- Windows: SetPriorityClass

- **Oczekiwanie na zakończenie procesu:**

- Linux: wait, waitpid
- Windows: brak bezpośredniego wsparcia, należy użyć odpowiednich mechanizmów synchronizacji

#### **14. Stany procesu.**

Proces po utworzeniu znajduje się w stanie SIDL. Stan ten ulega zmianie na SRUN, gdy zostaną przydzielone wystarczające zasoby do rozpoczęcia wykonywania procesu. Od tej chwili stan procesu oscyluje między SRUN, SSLEEP i SSTOP, dopóki proces się nie zakończy. Proces zakończony znajduje się w stanie SZOMB do czasu powiadomienia procesu macierzystego o tym fakcie. Stany procesu przedstawia

**SIDL** – stan pośredni podczas tworzenia procesu,

**SRUN** – wykonywalny,

**SSLEEP** – oczekujący (na zdarzenie),

**SSTOP** – zawieszony (proces jest zatrzymany przez sygnał lub proces macierzysty),

**SZOMB** – stan pośredni podczas usuwania procesu; proces w tym stanie nazywamy procesem uśpionym

#### **15. Komunikacja między procesami.**

W celu komunikacji między procesami w systemie UNIX wykorzystuje się:

- przekazywanie komunikatów
- pamięć dzielona - najszybsza metoda komunikacji między procesowej
- łącza komunikacyjne - Najprostszy sposób przekazywania danych między procesami. Są 2 rodzaje łączy: z kolejką FIFO i nienazwane

- Łącza nazwane (FIFO) - służą do komunikacji na zasadzie kolejki prostej. Kolejka FIFO umożliwia wymianę informacji pomiędzy procesami nieznanymi ze sobą
- Sygnały
- Pliki
- Semafora

## **16. Synchronizacja procesów.**

Problem synchronizacji procesów pojawia się wszędzie tam, gdzie mamy do czynienia ze współpracującymi ze sobą współbieżnymi procesami. Oto najczęściej spotykane przyczyny, dla których konieczna jest synchronizacja współpracujących procesów:

- Procesy współdzielą pewną strukturę danych -- Zwykle wykonania operacji na tej strukturze danych nie mogą dowolnie przeplatać się współbieżnie ze sobą. Pojedyncze operacje muszą być "zatomizowane", tzn. tylko jeden proces może na raz modyfikować współdzieloną strukturę danych, stąd konieczność synchronizacji procesów przy dostępie do struktury danych. Jest to przykład tzw. problemu sekcji krytycznej.
- Wyniki działania jednego procesu stanowią dane dla innego procesu. Oczywiście drugi z procesów może przetwarzać dane dopiero wówczas, gdy zostaną one obliczone przez pierwszy z procesów, stąd konieczność synchronizacji działań obydwu procesów. Dodatkowo, jeżeli dane zajmują dużo pamięci i są obliczane stopniowo, to drugi z procesów może informować pierwszy, które dane zostały "zużyte" i można wykorzystać zajmowaną przez nie pamięć. Przykładem takiego współdziałania procesów jest problem producenta-konsumenta.
- Procesy korzystają z pewnej wspólnej puli zasobów, które pobierają i zwalniają wedle potrzeb. Oczekiwanie na zasoby i przydzielanie ich wymaga również synchronizacji między procesami. Przykładem dobrze ilustrującym taki rodzaj synchronizacji jest problem pięciu filozofów.

## **WERSJA OD STASI**

### **Synchronizacja procesów producent - konsument**

- W systemie pracuje  $P (P \geq 1)$  procesów producenta i  $K (K \geq 1)$  procesów konsumenta. Każdy proces producenta przygotowuje porcję informacji, a następnie przekazuje ją procesowi konsumenta.
- Procesy producenta i konsumenta muszą podlegać synchronizacji, aby konsument nie próbował konsumować tych jednostek, które nie zostały jeszcze wyprodukowane.

## Synchronizacja procesów czytelnicy i pisarze

Dwie grupy procesów silnie konkurujących o zasoby:

- piszący
- czytający

**piszący** - muszą mieć zapewnione wykluczenie wzajemne względem siebie oraz względem procesów czytających przy korzystaniu z zasobu,

**czytający** - wiele procesów może jednocześnie być w posiadaniu zasobu, przy czym nie może z niego wtedy korzystać żaden z procesów piszących.

## Synchronizacja procesów pięciu filozofów

- Pięciu filozofów spędza życie na myśleniu i jedzeniu.
- Filozofowie dzielą wspólny okrągły stół, wokół którego ustawiono pięć krzeseł – po jednym dla każdego filozofa.
- Na środku stołu stoi misa ryżu, a naokoło leży pięć pałeczek każdego filozofa.
- Każdy filozof myśli, ale gdy zgłodnieje wówczas próbuje ująć w ręce dwie pałeczki leżące najbliżej jego miejsca przy stole.
- Za każdym razem filozof może podnieść tylko jedną pałeczkę.
- Kiedy filozof zdobędzie obie pałeczki, wtedy rozpocznie jedzenie nie rozstając się z pałeczkami ani na chwilę.
- Po zakończeniu jedzenia filozof odkłada obie pałeczki na stół

## Mechanizmy IPC

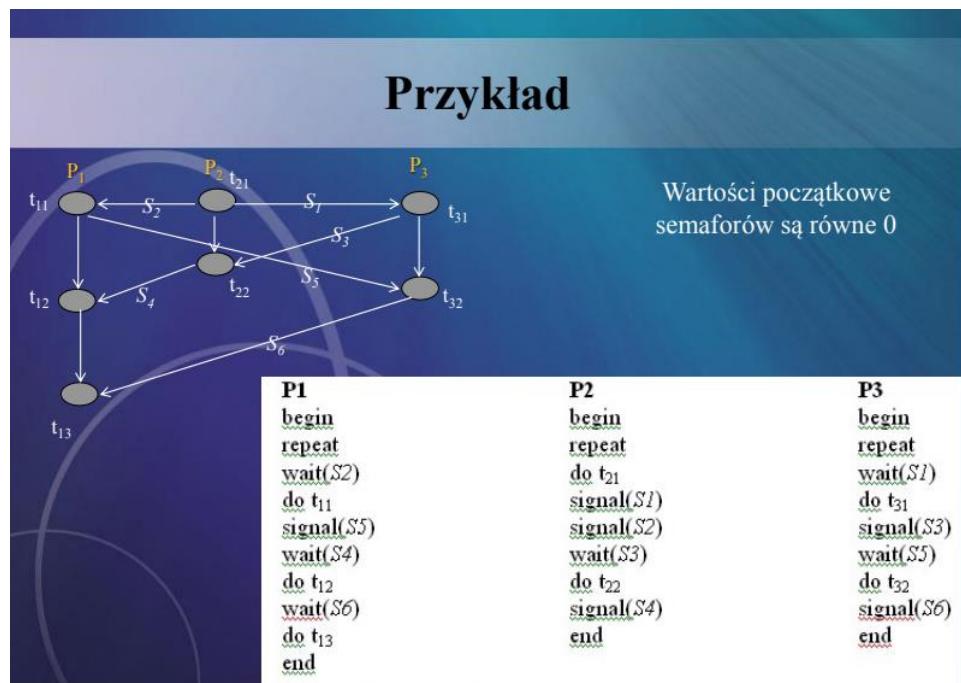
**Mechanizmy IPC (Inter Process Communication) to grupa mechanizmów komunikacji i synchronizacji procesów działających w ramach tego samego systemu operacyjnego**

Mechanizmy IPC obejmują:

- **Kolejki komunikatów** — umożliwiają przekazywanie określonych porcji danych.
- **Pamięć współdzieloną** — umożliwiają współdzielenie kilku procesom tego samego fragmentu wirtualnej przestrzeni adresowej,
- **Semafora** — umożliwiają synchronizację procesów w dostępie do współdzielonych zasobów np. do pamięci współdzielonej

### 17. Operacje semaforowe.

**Prostsza wersja:**



**Pojebana wersja:**

**OPERACJE SEMAFOROWE** Podstawowym problemem odnoszącym się do współpracy procesów jest wykluczanie wzajemne, rozumiane jako zapewnienie wyłączności działania w pewnym obszarze, nazywanym **sekcją krytyczną**, jednemu procesowi.

Punktem zwrotnym w rozwoju teorii systemów operacyjnych i przetwarzania równoległego , kiedy to Dijkstra'e przedstawił rozwiązanie problemu wykluczania, wprowadzając pojęcia semaforów oraz wykonywanych na nich operacji czekaj (**wait**) i sygnalizuj (**signal**). Podstawową ideą mechanizmu

semaforowego jest założenie, że procesy nie mogące wejść do sekcji krytycznej są zawieszane (wait), a po zwolnieniu jej przez inny proces są reaktywowane (**signal**). Zasady wchodzenia do sekcji krytycznej są następujące:

1. Jeśli sekcja jest wolna, proces wchodzi do niej bezpośrednio.
2. Jeśli sekcja jest zajęta, to proces jest zawieszany i oczekuje na jej zwolnienie.
3. Po zwolnieniu sekcji krytycznej proces sprawdza, czy żaden inny proces nie jest zawieszony w oczekiwaniu na wejście do niej, jeśli tak, to jednemu z nich umożliwia wejście do sekcji.

Aby zapewnić taki dostęp do sekcji krytycznej, należy związać z nią pewną zmienną typu semafor. Semafor jest to nieujemna liczba całkowita, na której z wyjątkiem nadawania wartości początkowych, można wykonywać jedynie operacje czekaj (określaną również mianem **wait** lub **P**) i sygnalizuj (określaną również mianem **signal** lub **V**), które są niepodzielne – nic nie może przerwać ich wykonania.

**Operacja sygnalizuj (signal, V)** odpowiada podniesieniu semafora. W jej wyniku następuje zwiększenie wartości semafora s o 1, przy czym traktuje się tę operację jako niepodzielną. Z jej niepodzielności wynika, że wykonanie operacji sygnalizuj nie jest równoważne wykonaniu instrukcji przypisania s: =s+1. Operacja ta jest wykonywana po wyjściu procesu z sekcji krytycznej:

**Operacja czekaj (wait, P)** odpowiada oczekiwaniu na podniesienie semafora, a następnie jego opuszczenie. W jej wyniku następuje zmniejszenie wartości semafora s o 1, o ile wartość ta jest dodatnia. Jeśli nie jest to możliwe (s=0), proces wywołujący tę operację jest zawieszany i będzie mógł być nadal wykonywany tylko wówczas, gdy inny proces zwiększy wartość semafora o 1 w wyniku operacji sygnalizuj. Również i ta operacja jest niepodzielna. Jeżeli kilka procesów było

## **18. Blokady.**

**Blokada**, jeżeli kilka procesów współzawodniczy o zasoby, to może wytworzyć się taka sytuacja, w której każdy proces aby móc nadal działać będzie musiał skorzystać z tych zasobów, których używa inny proces; wówczas żaden proces nie będzie mógł kontynuować swej pracy. Taką sytuację nazywamy blokadą lub zakleszczeniem (ang. deadlock). Można to rozumieć jako stan systemu, w

którym procesy pozostają zawieszone w oczekiwaniu na zdarzenia, które nigdy nie zajdą. Jednym z zadań systemu operacyjnego jest niedopuszczanie do powstania blokady albo przynajmniej ograniczanie jej skutków.

### 19. Zadania planistów.

- Planista średioterminowy odpowiedzialny jest za wymianę (swapping) procesów między pamięcią operacyjną a dyskiem – stosowany często w systemach z podziałem czasu.

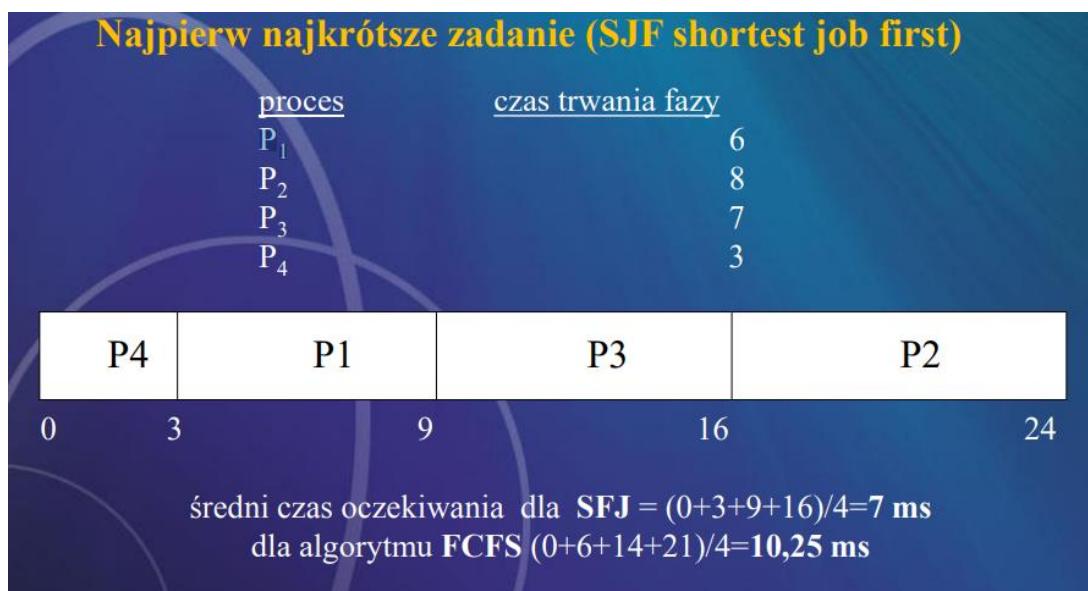
Zadanie planisty długoterminowego - dobranie dobrej mieszanki procesów (process mix) zawierającej procesy obydwu rodzajów

Planista krótko terminowy odpowiedzialny jest za przejście ze stanu gotowości do wykonania

### 20. Algorytmy planowania przydziału procesora.

Planowanie metodą FCFS (first come first served)

- Proces, który pierwszy zamówił procesor pierwszy go otrzyma.
- Implementacja - za pomocą kolejki FIFO.
- Blok kontrolny procesu wchodzącego do kolejki procesów gotowych jest dodawany do końca.
- Wolny procesor przydziela się procesowi z czoła kolejki procesów gotowych.
- Średni czas oczekiwania bywa bardzo długi.



-FCFS (First Come First Served) — pierwszy zgłoszony, pierwszy obsłużony - Procesy otrzymują procesor w kolejności, w jakiej zgłosiły się do systemu. Specyficzną cechą w systemach komputerowych jest możliwość oddania procesora innemu procesowi na czas oczekiwania na przydział dodatkowego zasobu

-LCFS (Last Come First Served) — ostatni zgłoszony, pierwszy obsłużony - obsługuje procesy w kolejności odwrotnej do kolejności zgłoszeń. Algorytm nie wywala procesów, więc nowo

przychodzący proces jest pierwszy w kolejce i czeka na zwolnienie procesora przez bieżąco wykonywany proces

-SJF (Shortest Job First) – najpierw najkrótsze zadanie - preferuje procesy, które mają najmniejsze wymagania odnośnie czasu procesora, potrzebnego na realizację przetwarzania.

## **21. Zarządzanie pamięcią operacyjną w systemach LINUX, WINDOWS.**

### **LINUX:**

- adres logiczny - tworzony przez procesor (adres wirtualny)

\*Jądro Linux prowadzi rejestr używanej i nieużywanej pamięci.

- odwzorowanie adresów wirtualnych na fizyczne - MMU (jednostka zarządzająca pamięcią – memory management unit)

### **WINDOWS:**

- Domyślny rozmiar procesu w 32-bitowym systemie Windows wynosi 2 GB.
- Jeśli plik wykonywalny jest oznaczony jako wymagający dużej przestrzeni adresowej proces 32-bitowy może osiągnąć 3 GB w systemie 32-bitowym lub 4 GB w systemie 64-bitowym.
- Wirtualna przestrzeń adresowa procesu w 64-bitowym systemie Windows 8 a także w Windows Server 2012 zajmowała 8 TB, a w 64-bitowym Windows od wersji 8.1 wynosi 128 TB.
- Maksymalny rozmiar pamięci fizycznej obsługiwanej przez system Windows waha się od 2 GB do 24 TB w zależności od wersji i edycji systemu.

## **22. Pamięć wirtualna – organizacja pamięci - implementacja pamięci wirtualnej.**

Pamięć wirtualna jest techniką programową a także sprzętową gospodarowania pamięcią operacyjną RAM pozwalającą na przydzielanie pamięci dla wielu procesów, zwalnianie jej i powtórne przydzielanie, w ilości większej niż rzeczywista ilość pamięci fizycznej zainstalowanej w komputerze poprzez przeniesienie danych z ostatnio nie używanej pamięci do pamięci masowej (np. twardego dysku); w sytuacji gdy procesor odwołuje się do danych z pamięci przeniesionej na dysk przesuwa się te dane do pamięci w wolne miejsce, a gdy brak wolnej pamięci - zwalnia się ją przez wyżej opisane przerzucenie jej na dysk

## **23. Stronicowanie pamięci.**

Prawie wszystkie istniejące obecnie implementacje dzielą wirtualną przestrzeń adresową procesu na strony. Strona to obszar ciągłej pamięci o stałym rozmiarze (zazwyczaj 4kB). Systemy, gdzie

zapotrzebowanie na wielkość wirtualnej przestrzeni adresowej jest większe lub dysponujące większymi zasobami pamięci operacyjnej mogą używać stron o większym rozmiarze.

- Adres logiczny generowany przez proces składa się z dwóch części:
- numeru strony s (page number) – indeks w tablicy stron (tablica stron zawiera adresy bazowe wszystkich stron w pamięci operacyjnej).
- odległość na stronie o (page offset) – w połączeniu z adresem bazowym daje adres fizyczny pamięci, który jest wysyłany do jednostki pamięci.

#### **24. Zadania mechanizmu stronicowania.**

- odwzorowywanie adresów wirtualnych w adresy rzeczywiste:
  1. określenie, do której strony odnosi się adres w programie,
  2. znalezienie ramy, którą aktualnie zajmuje dana strona.
- przesyłanie - w zależności od potrzeby - stron z pamięci pomocniczej do pamięci operacyjnej oraz odsyłanie już nieużywanych stron z powrotem.

#### **25. Segmentacja pamięci.**

- Segmentacja jest schematem zarządzania pamięcią urzeczywistniającym sposób widzenia pamięci przez użytkownika.
- Przestrzeń adresów logicznych jest zbiorem segmentów.
  - Segment to jednostka logiczna, taka jak: program główny, procedury, funkcje, metody, stosy, wykazy, tablice, zmienne itd.
  - Dla ułatwienia implementacji segmenty są ponumerowane.
- Adres logiczny:<numer segmentu, odległość>
- Tablica segmentów: odwzorowuje dwuwymiarowe adresy logiczne w jednowymiarowe adresy fizyczne. Każda pozycja składa się z dwu części:
  - Baza segmentu – zawiera początkowy adres fizyczny segmentu w pamięci
  - Granica segmentu - określa długość segmentu

Jedna z metod ochrony pamięci, używana przy wielozadaniowości. Każdy proces otrzymuje swój własny obszar pamięci, realizowany poprzez rejesty segmentowe. polega na podzieleniu przez procesor pamięci fizycznej na ciągłe bloki nazywane segmentami, które opisane są przez deskryptory zawierające:

- adres bazowy — fizyczny adres początku segmentu w pamięci
- rozmiar — długość segmentu w ustalonych jednostkach
- atrybuty określające rodzaj zawartości i dostępność (np. kod programu)
- identyfikator (określany też jako nazwa) — wartość wskazująca na opis segmentu w tablicy segmentów

## **26. Porównanie stronicowania i segmentacji.**

- celem stronicowania jest fizyczny podział pamięci; celem segmentacji jest logiczny podział pamięci operacyjnej
- stronicowanie jest mechanizmem niskiego poziomu, niewidocznym dla programisty; segmentacja jest mechanizmem wyższego poziomu, widocznym dla programisty,
- rozmiar strony stały, ustalona, wynikający z architektury; rozmiar segmentu dowolny określany przez programistę,
- zarówno w przypadku zastosowania stronicowania jak i segmentacji całkowita przestrzeń adresowa może być większa od dostępnej pamięci fizycznej,
- segmentacja umożliwia lepszą ochronę poszczególnych elementów procesu poprzez możliwość rozróżniania segmentów logicznie grupujących elementy procesów,

## **27. Strategie przydziału pamięci.**

- Przydział ciągły - Polega na tym, że każdemu procesorowi przydziela się jeden spójny fragment pamięci. Pamięć jest podzielona na zajęte i wolne fragmenty, ponieważ aby system funkcjonował muszą działać jakieś procesy. Proces posiada adresy fizyczne od początku przydzielonego obszaru, do końca - 1, przez co nie wyjdzie on poza zakres swojej pamięci.
- Stronicowanie - Proces widzi spójny obszar pamięci logicznej, ale nie tworzy spójnego obszaru w pamięci fizycznej. Program widzi poszczególne strony, których wielkość jest określona potągią dwójką np. 512b, 4kB itd. Program porusza się po pamięci za pomocą stron.
- Segmentacja - Podzielenie pamięci na poszczególne segmenty, np. kod, zmienne, stos, sterta, różni się od przydziału ciągłego tym, że program ma dostęp do kilku segmentów zamiast do jednego.
- Segmentacja ze stronicowaniem - Połączenie stronicowania i segmentacji.

Proces widzi pamięć logiczną składającą się z szeregu segmentów. Poszczególne segmenty nie są rozmieszczone spójnie, a są podzielone na strony i każda strona może się znajdować w dowolnej ramce. Adres logiczny składa się z numeru segmentu i adresu w obrębie segmentu. Na podstawie numeru segmentu można wyznaczyć tablicę stron określającą rozmieszczenie stron tworzących segment.

## **28. Strategie rozmieszczania dla systemów bez stronicowania.**

System prowadzi ewidencję dziur, wiadomo ile jest dziur i jakie są duże. Wyróżniamy następujące strategie rozmieszczenia:

1. Najlepsze dopasowanie:
  - Dziury są uporządkowane malejąco pod względem rozmiaru,
  - System umieszcza proces w pierwszej dziurze na liście.
2. Najgorsze dopasowanie:
  - Dziury są uporządkowane rosnąco pod względem ich adresów bazowych,
  - System przydziela pierwszą z możliwych dziur.
3. Pierwsze dopasowanie:

- Dziury są uporządkowane rosnąco pod względem ich adresów bazowych,
  - System przydziela pierwszą z możliwych dziur.
4. Algorytm bliźniaków:
- Rozmiar dziur jest zawsze  $2^n$
  - Gdy potrzebna ilość pamięci jest mniejsza niż połowa dziury, system dzieli wybraną dziurę na dwie równe części do momentu aż dziura będzie odpowiednia. Blok umieszczany jest w powstałej dziurze.
  - Jeśli nie ma wystarczająco dużej dziury system łączy dwie takie same sąsiadujące dziury w jedną dwa razy większą.

## Dla ubogich

Sposoby:

-First fit

Wybierany jest pierwszy wolny obszar, który ma wystarczającą ilość miejsca

-Best fit

Wybierany jest najmniejszy wolny obszar, który ma wystarczającą ilość miejsca

-Worst fit

Wybierany jest największy wolny obszar, który ma wystarczającą ilość miejsca

## **29. Strategie rozmieszczania dla systemów ze stronicowaniem.**

W systemach ze stronicowaniem strategia rozmieszczania polega na wyborze wymaganej ilości wolnych ramek. Problem fragmentacji pamięci nie istnieje, ze względu na stały rozmiar ramek.

- Nie ma potrzeby przechowywania listy dziur,
- Występuje lista wolnych stron,
- Nie powoduje fragmentacji zewnętrznej
- Występuje fragmentacja wewnętrzna (bloki o stałej wielkości są przydzielane do procesu bez uwzględniania rozmiaru procesu).

## **30. Strategie wymiany dla systemów ze stronicowaniem.**

Proces żąda dostępu do strony w obszarze wymiany. W przypadku braku wolnych ramek na dysk jest odsyłana strona:

1. Najdawniej używana:
  - w deskryptorze (identyfikator pliku wykorzystywany przez system operacyjny) strony musi być zapisany czas odwołania do strony,
  - stosowanie tego algorytmu jest obciążone kosztami zapisywania czasu odwołań do wszystkich stron.
2. Najmniej używana:

- w tym algorytmie należy odnotowywaćczęstość użycia strony,
  - wadą tej metody jest fakt, że ostatnio załadowana strona może być nieroźważnie wymieniona,
  - można zabronić wymiany stron załadowanych podczas określonego przedziału czasu.
3. Najdawniej załadowana:
- algorytm ten wymaga zapisu ciągu stron ładowanych do pamięci, co pozwala wskazać najdawniej załadowane strony,
  - pomija się w nim fakt, że najwięcej odniesień może być do strony, która najdłużej znajduje się w pamięci.

## Algorytm FIFO zastępowania stron

Przyjrzyjmy się teraz, co się stanie, gdy liczba ramek wyniesie 4

| Chwila    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|
| Odwołanie | 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3  | 4  | 5  |
| Ramka 1   | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5  | 4  | 4  |
| Ramka 2   |   | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1  | 1  | 5  |
| Ramka 3   |   |   | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2  | 2  | 2  |
| Ramka 4   |   |   |   | 4 | 4 | 4 | 4 | 4 | 4 | 3  | 3  | 3  |

ofiarami staje się strona, która najdłużej przebywa w pamięci

## Algorytm optymalny

| Chwila    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|
| Odwołanie | 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3  | 4  | 5  |
| Ramka 1   | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 4  | 4  |
| Ramka 2   |   | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2  | 2  | 2  |
| Ramka 3   |   |   | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  |
| Ramka 4   |   |   |   | 4 | 4 | 4 | 5 | 5 | 5 | 5  | 5  | 5  |

ofiarami staje się strona, która będzie nieużywana przez najdłuższy okres

## Algorytm LRU

| Chwila    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|
| Odwółanie | 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3  | 4  | 5  |
| Ramka 1   | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 5  |
| Ramka 2   |   | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2  | 2  | 2  |
| Ramka 3   |   |   | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 5  | 4  | 4  |
| Ramka 4   |   |   |   | 4 | 4 | 4 | 4 | 4 | 4 | 3  | 3  | 3  |

ofiarą strona, która nie była używana od najbliższego czasu

### 31. Strategie pobierania z pamięci pomocniczej.

Strategie pobierania służą do określania momentu, w którym bloki mają być załadowane do pamięci operacyjnej z obszaru wymiany, z pamięci pomocniczej:

1. Pobieranie na żądanie:
  - wystąpienie błędu braku bloku (segmentu lub strony) generuje żądanie pobrania bloku, a strategie rozmieszczania służą do przydziału pamięci dla nowego bloku,
  - w systemach bez stronicowania przesyła się bloki najczęściej na żądanie.
2. Pobieranie przewidujące:
  - System może pobierać nowe bloki w oparciu o:
    - znajomość konstrukcji programu
    - wnioskowanie z dotychczasowego działania procesu

### 32. Zarządzanie pamięcią w systemach LINUX oraz WINDOWS.

### 33. Systemy we/wy.

**Systemy wejścia-wyjścia** - mają na celu umożliwienie wprowadzania informacji do komputera (wpisywanie tekstu, odczytywanie i zapisywanie danych na urządzeniach pamięci masowej itp.) oraz jego wyprowadzanie (np. wyświetlanie na monitorze, drukowanie)

**Sprzęt wejścia-wyjścia:**

- pamięci masowe (dyski, taśmy)
- urządzenia komunikacji z użytkownikiem (monitory, klawiatury, myszki),
- urządzenie komunikacyjne (karty sieciowe, modemy)
- inne urządzenia specjalistyczne (kierownice, manetki, ...)

Urządzenia WE/WY pośredniczą w wymianie informacji między procesorem a układami zewnętrznymi. Korzystają z interfejsów, czyli ustalonych reguł i środków technicznych służących do połączenia komputera z urządzeniami zewnętrznymi. Interfejs składa się z :

- Wymagań logicznych
- Wymagań elektrycznych
- Wymagań konstrukcyjnych

Interfejs pozwala przesyłać dane równolegle oraz szeregowo w obu kierunkach jak i w jednym kierunku w zależności od potrzeb.

Dla procesora urządzenie WE/WY jest widoczne jako rejestr lub kilka rejestrów o określonych adresach i zestaw sygnałów sterujących.

#### **34. Podstawowe własności systemów rozproszonych.**

- dzielenie zasobów
- otwartość
- współbieżność
- skalowalność
- tolerowanie uszkodzeń
- przeźroczystość

#### **35. Model klient-serwer.**

- Podział systemu operacyjnego na moduły
- Moduły nie są rozmieszczone w warstwach
- Nie komunikują się między sobą poprzez wywoływanie procedur, ale wysyłają komunikaty za pośrednictwem centralnego programu obsługi komunikatów
- Komunikaty mogą być wysyłanie w obie strony
- Moduł wysyłający początkowy komunikat nazywa się **klientem**
- Moduł odbierający ten komunikat **serwerem**

#### **36. Technologia serwerowa i jej usługi.**

#### **37. Synchronizacja w systemach rozproszonych.**

- Synchronizacja w systemach rozproszonych jest procesem bardziej złożonym niż w systemach skentralizowanych.
- W zagadnieniach synchronizacji procesów na wielu maszynach nie jest istotna znajomość czasu absolutnego - istotne jest, aby procesy mogły ustalić kolejność zdarzeń.
- Powodu pomiar czasu możemy podzielić na:
  - pomiar czasu fizyczny
  - zegary wskazują czas astronomiczny,
  - pomiar czasu logiczny
  - zegary o wzajemnie uzgodnionym czasie, niekoniecznie astronomicznym.

### **38. Procesy w systemach rozproszonych.**

Proces rozproszony - powinien być niewrażliwy na awarie poszczególnych maszyn . Proces rozproszony jest definiowany jako pewien zbiór procesów sekwencyjnych, czyli takich które reprezentują wykonanie pewnego ciągu operacji. Dodatkowo procesy mogą być współbieżne, a ich działanie jest skoordynowane, tak aby mogły zrealizować pewien wspólny cel.

Wyróżnia się dwa modele rozproszonych procesów:

- model statyczny (ang. static process model )
- model dynamiczny (ang. dynamic process model ).

W modelu statycznym zakłada się, że procesy, które tworzą pewne zadanie, tworzone są na początku jego działania. Drugi model nie nakłada takiego ograniczenia i procesy można tworzyć i niszczyć w dowolnym momencie trwania programu – zadania.

### **39. Rozproszona pamięć dzielona.**

**Rozproszona pamięć dzielona** (ang. distributed shared memory, w skrócie DSM) jest abstrakcją używaną do określenia wspólnego użytkowania danych przez procesy, które nie dzielą pamięci fizycznej. Pamięć dzielona stanowi część przestrzeni adresowej procesów i mogą one z niej korzystać w standardowy sposób. Zadaniem systemu operacyjnego ew. systemu wykonawczego jest zapewnienie w sposób przezroczysty dostępu do wspólnych danych (emulowanie wspólnej pamięci fizycznej).

Główym celem DSM jest zapewnienie programiście wygodnego dostępu do współdzielonych danych, bez konieczności korzystania z mechanizmu przekazywania komunikatów. Oczywiście w systemie rozproszonym nie da się całkowicie uniknąć przekazywania komunikatów - korzysta z niego system wsparcia działania pamięci DSM wysyłając aktualizacje między komputerami.

Przekazywanie komunikatów a pamięć DSM:

- Model programowania
- Wydajność

Podstawowe podejścia do realizacji pamięci DSM:

- Wsparcie przez stronicowanie
- Wsparcie biblioteczne

Struktura danych przechowywanych w pamięci DSM:

- Obraz bajtowy
- Obiekty dzielone
- Dane stałe

**40. Bezpieczeństwo i ochrona.**

**ZAGADNIENIA DO EGZAMINU – INFORMATYKA ( MOODLE )**

## 1.2. Zagadnienia – 15.

### 1. Architektura, zadania właściwości systemu operacyjnego LINUX oraz WINDOWS.

### 2. Organizacja pamięci pomocniczej.

- Pamięć pomocnicza jest zarządzana przez system plików.
- System operacyjny dostarcza jednolitego logicznie obrazu przechowywania informacji w oderwaniu od cech fizycznych urządzeń.
- PLIK - logiczna jednostka informacji.
- System plikowy:
  - zbiór plików
  - struktura katalogów

### 3. Systemy plików LINUX oraz WINDOWS.

System plików - metoda przechowywania plików, zarządzania plikami, informacjami o tych plikach, tak by dostęp do plików i danych w nich zgromadzonych był łatwy dla użytkownika systemu; także: wolumin.

#### Linux:

- Ext (ang. *Extended File System*) – pierwsza wersja najbardziej popularnego linuksowego systemu plików. Dopuszczał on pliki o rozmiarze do 64 MB i partycje do 2 GB. Poważną niedogodnością ext był fakt, że po dłuższym działaniu prowadził do dużej fragmentacji. Został szybko zastąpiony przez ext2.
- Ext2 (ang. *Second Extended File System*) – drugi rozszerzony system plików dla systemu Linux. System plików ext2 zawiera mechanizm zapobiegający znacznej fragmentacji danych, co zdarzało się podczas używania poprzedniej jego wersji. Ext2 przy domyślnym rozmiarze bloku (4 KB) obsługuje partycje o wielkości do 4 TB i pojedyncze pliki o wielkości do 2 GB.
- Ext3 (ang. *Third Extended File System*) – system plików oparty na systemie ext2. System plików ext3 jest rozszerzeniem ext2 i różni się od niego dodanym mechanizmem księgowania, który w razie awarii systemu umożliwia szybsze przywrócenie spójności systemu plików niż w przypadku ext2. Poza mechanizmem księgowania system plików ext3 różni się od ext2 brakiem możliwości odzyskania skasowanych plików.
- Ext4 (ang. *Fourth Extended File System*) – czwarta wersja rozszerzonego systemu plików, następcą ext3. Umożliwia obsługę woluminów do 1 eksbibajta (EiB). Wielkość pojedynczego pliku nie może przekraczać 16 tebibajtów (TiB). W systemie ext4 pliki przechowywane są w ciągłym zbiorze bloków, nazywanym extent.
- ReiserFS (zwany także Reiser3) – system plików będący jednym z pierwszych systemów plików z księgowaniem dla Linuksa. Podobnie jak w ext3, księgowanie zapewnia atomowość operacji na systemie plików w odróżnieniu od ext2, dla którego czas reakcji systemu plików jest nieprzewidywalny. Za zwiększenie bezpieczeństwa danych płaci się niską szybkością operacji i przestrzenią dyskową.
- XFS – 64-bitowy system plików. XFS pozwala na obsługę dużych dysków twardych. Maksymalny rozmiar woluminu jest ograniczony do 16 milionów TB. Natomiast rozmiar pojedynczego pliku może wynosić maksymalnie 263 bajtów czyli ponad 8 milionów TB (dokładnie 8 388 608 TB).

#### Windows:

- File Allocation Table (FAT) – system plików zastosowany w systemach operacyjnych takich jak DOS i Windows. FAT to lista klastrów, która pokazuje, gdzie przechowywany jest plik komputerowy. Klastry te niekoniecznie znajdują się tuż obok siebie, ale mogą być rozproszone na dysku lub nośniku pamięci. FAT znajduje się zawsze w pierwszych kilku klastrach urządzenia pamięci masowej.
- NTFS (ang. *New Technology File System*) – standardowy system plików systemu Windows NT i jego następców. NTFS może obsługiwać dyski twarde nawet do 16 EB. Pojedynczy rozmiar pliku ograniczony jest do mniej niż 256 TB. Encrypting File System (EFS) to funkcja NTFS, zapewniająca szyfrowanie na poziomie plików, co oznacza, że można szyfrować pojedyncze pliki i foldery. NTFS jest dziennikowym systemem plików, co oznacza, że zmiany systemu są zapisywane w dzienniku, zanim zostaną one faktycznie zapisane.
- REFS (Resilient File System) – system plików będący następcą NTFS. Sprawdza się w sytuacji gdzie potrzeba przetwarzać duże ilości danych/plików oraz przy ich udostępnianiu lub współdzieleniu (serwer plików). REFS potrafi obsłużyć dyski o pojemności 1 jottabajta (YB) – 1024 bajta. Systemu plików potrafi rozpoznać problem z danymi i naprawić je samodzielnie.

## **4. Zarządzanie procesami i wątkami w systemach LINUX oraz WINDOWS**

- **Komunikacja między procesami.**

- Dla funkcjonowania systemów rozproszonych konieczna jest sprawna komunikacja pomiędzy odległymi procesami.
- Podstawowym mechanizmem komunikacji międzyprocesowej jest wymiana komunikatów między procesami.
- Wymiana komunikatów polega na istnieniu pary operacji wyslij/odbierz wywoływanych odpowiednio przez nadawcę i odbiorcę komunikatu

## Komunikacja międzyprocesowa

• Procesy *i*, *j* oraz *k* wysłały wiadomości A-G.

• Wiadomości A, B, D i G dotarły do miejsca przeznaczenia, czyli innego procesu.

• Komunikaty C, E i F są ciągle w drodze kiedy procesor spowodował błąd procesu *i*.

• W początkowej próbie naprawy, przy użyciu ostatniego punktu kontrolnego (zazначенego jako "["), można odtworzyć stany procesów A, B i G, ponieważ początek i koniec komunikatu zostały zapisane (zapamiętane), a komunikat G jeszcze się nie rozpoczął, co zwalnia od potrzeby jego odzyskania.

• Komunikaty A i B nazywamy *normalnymi*, natomiast komunikat G określamy jako *zagubionymi*.

Proces *i*  
Proces *j*  
Proces *k*

A-G - komunikaty [ - punkt kontrolny \* - awaria

autor: dr inż. Stanisława Plichta

## Komunikacja międzyprocesowa

- Wiadomości E i F są tak zwany komunikatami sierotami i stwarzają najpoważniejszy problem.
- Gdy dotrą one do swoich celów, zostaną odrzucone ponieważ wiadomości o ich wysłaniu nie zostały zatwierdzone.
- Procesy podczas procesu rekonstrukcji od ostatnich punktów kontrolnych, wygenerują od nowa te komunikaty.
- W czasie procesu odbudowy może dojść do podwojenia ilości tych samych komunikatów.

Proces *i*  
Proces *j*  
Proces *k*

D, M - komunikaty [ - punkt kontrolny \* - awaria ] - punkt odbudowy

utor: dr inż. Stanisława Plichta

Cechy komunikacji międzyprocesowej

- Różne realizacje techniki wymiany komunikatów charakteryzują się różnymi właściwościami, które dotyczą:
  - sposobu adresowania
  - formatu danych przekazywanych w komunikatach
  - sposobu synchronizacji operacji send i receive:
    - każda operacja może być blokująca lub nieblokująca
    - w ramach konkretnej realizacji można ustalić precyzyjnie, kiedy następuje powrót z określonej procedury.
  - niezawodność – gwarancje dostarczenia komunikatu
  - uporządkowanie – kolejność dostarczania komunikatów w odniesieniu do kolejności nadawania komunikatów.

- **Synchronizacja procesów.**

## **Synchronizacja procesów producent - konsument**

- W systemie pracuje  $P (P \geq 1)$  procesów producenta i  $K (K \geq 1)$  procesów konsumenta. Każdy proces producenta przygotowuje porcję informacji, a następnie przekazuje ją procesowi konsumenta.
- Procesy producenta i konsumenta muszą podlegać synchronizacji, aby konsument nie próbował konsumować tych jednostek, które nie zostały jeszcze wyprodukowane.

## Synchronizacja procesów czytelnicy i pisarze

Dwie grupy procesów silnie konkurujących o zasoby:

- piszący
- czytający

**piszący** - muszą mieć zapewnione wykluczenie wzajemne względem siebie oraz względem procesów czytających przy korzystaniu z zasobu,

**czytający** - wiele procesów może jednocześnie być w posiadaniu zasobu, przy czym nie może z niego wtedy korzystać żaden z procesów piszących.

## Synchronizacja procesów pięciu filozofów

- Pięciu filozofów spędza życie na myśleniu i jedzeniu.
- Filozofowie dzielą wspólny okrągły stół, wokół którego ustawiono pięć krzeseł – po jednym dla każdego filozofa.
- Na środku stołu stoi misa ryżu, a naokoło leży pięć pałeczek każdego filozofa.
- Każdy filozof myśli, ale gdy zgłodnieje wówczas próbuje ująć w ręce dwie pałeczki leżące najbliżej jego miejsca przy stole.
- Za każdym razem filozof może podnieść tylko jedną pałeczkę.
- Kiedy filozof zdobędzie obie pałeczki, wtedy rozpocznie jedzenie nie rozstając się z pałeczkami ani na chwilę.
- Po zakończeniu jedzenia filozof odkłada obie pałeczki na stół

## Mechanizmy IPC

**Mechanizmy IPC (Inter Process Communication) to grupa mechanizmów komunikacji i synchronizacji procesów działających w ramach tego samego systemu operacyjnego**

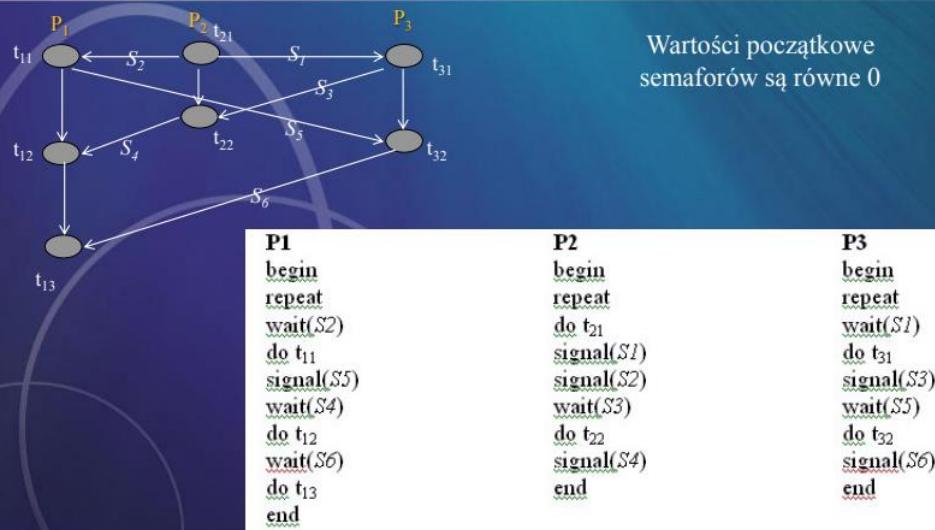
Mechanizmy IPC obejmują:

- **Kolejki komunikatów** — umożliwiają przekazywanie określonych porcji danych.
- **Pamięć współdzieloną** — umożliwiają współdzielenie kilku procesom tego samego fragmentu wirtualnej przestrzeni adresowej,
- **Semafora** — umożliwiają synchronizację procesów w dostępie do współdzielonych zasobów np. do pamięci współdzielonej

- Operacje semaforowe.

Prostsza wersja:

## Przykład



Pojebana wersja:

**OPERACJE SEMAFOROWE** Podstawowym problemem odnoszącym się do współpracy procesów jest wykluczanie wzajemne, rozumiane jako zapewnienie wyłączności działania w pewnym obszarze, nazywanym **sekcją krytyczną**, jednemu procesowi.

Punktem zwrotnym w rozwoju teorii systemów operacyjnych i przetwarzania równoległego , kiedy to Dijkstra'e przedstawił rozwiązanie problemu wykluczania, wprowadzając pojęcia semaforów oraz wykonywanych na nich operacji czekaj (**wait**) i sygnalizuj (**signal**). Podstawową ideą mechanizmu

semaforowego jest założenie, że procesy nie mogące wejść do sekcji krytycznej są zawieszane (wait), a po zwolnieniu jej przez inny proces są reaktywowane (**signal**). Zasady wchodzenia do sekcji krytycznej są następujące:

1. Jeśli sekcja jest wolna, proces wchodzi do niej bezpośrednio.
2. Jeśli sekcja jest zajęta, to proces jest zawieszany i oczekuje na jej zwolnienie.
3. Po zwolnieniu sekcji krytycznej proces sprawdza, czy żaden inny proces nie jest zawieszony w oczekiwaniu na wejście do niej, jeśli tak, to jednemu z nich umożliwia wejście do sekcji.

Aby zapewnić taki dostęp do sekcji krytycznej, należy związać z nią pewną zmienną typu semafor. Semafor jest to nieujemna liczba całkowita, na której z wyjątkiem nadawania wartości początkowych, można wykonywać jedynie operacje czekaj (określaną również mianem **wait** lub **P**) i sygnalizuj (określaną również mianem **signal** lub **V**), które są niepodzielne – nic nie może przerwać ich wykonania.

**Operacja sygnalizuj (signal, V)** odpowiada podniesieniu semafora. W jej wyniku następuje zwiększenie wartości semafora s o 1, przy czym traktuje się tę operację jako niepodzielną. Z jej niepodzielności wynika, że wykonanie operacji sygnalizuj nie jest równoważne wykonaniu instrukcji przypisania s: =s+1. Operacja ta jest wykonywana po wyjściu procesu z sekcji krytycznej:

**Operacja czekaj (wait, P)** odpowiada oczekiwaniu na podniesienie semafora, a następnie jego opuszczenie. W jej wyniku następuje zmniejszenie wartości semafora s o 1, o ile wartość ta jest dodatnia. Jeśli nie jest to możliwe (s=0), proces wywołujący tę operację jest zawieszany i będzie mógł być nadal wykonywany tylko wówczas, gdy inny proces zwiększy wartość semafora o 1 w wyniku operacji sygnalizuj. Również i ta operacja jest niepodzielna. Jeżeli kilka procesów było

- **Zadania planistów.**
  - Planista średnioterminowy odpowiedzialny jest za wymianę (swapping) procesów między pamięcią operacyjną a dyskiem – stosowany często w systemach z podziałem czasu.
  - Zadanie planisty długoterminowego - dobranie dobrej mieszanki procesów (process mix) zawierającej procesy obydwu rodzajów
  - Planista krótko terminowy odpowiedzialny jest za przejście ze stanu gotowości do wykonania

## Co za chuj to usunął

- Krótkoterminowy – planista przydziału procesora, który podejmuje decyzję, kiedy proces ma przejść ze stanu gotowego w wykonywany
- Średnioterminowy – odpowiada za decyzję, które procesy usunąć z pamięci, a które ponownie załadować.
- Długoterminowy – planista zadań, podejmuje decyzję o przyjęciu nowego procesu do systemu i kontrolą zadań w systemie oraz ich doborem do wykorzystania zasobów

- Algorytmu planowania przydziału procesora.

-FCFS (First Come First Served) — pierwszy zgłoszony, pierwszy obsłużony - Procesy otrzymują procesor w kolejności, w jakiej zgłosiły się do systemu. Specyficzną cechą w systemach komputerowych jest możliwość oddania procesora innemu procesowi na czas oczekiwania na przydział dodatkowego zasobu

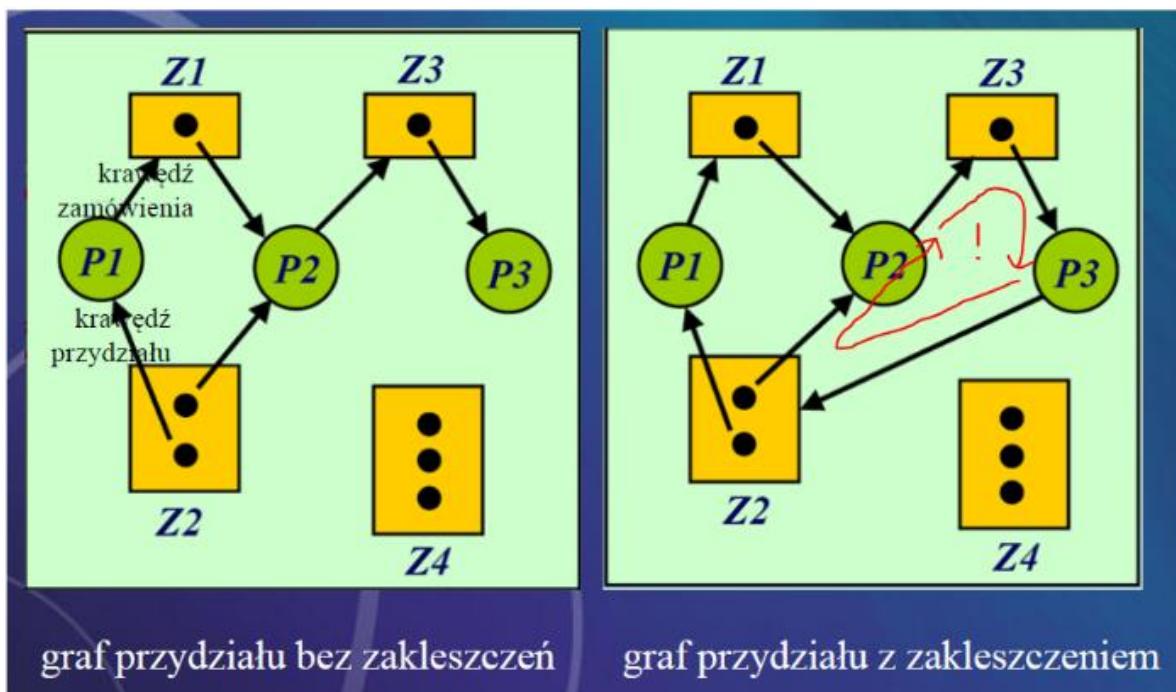
-LCFS (Last Come First Served) — ostatni zgłoszony, pierwszy obsłużony - obsługuje procesy w kolejności odwrotnej do kolejności zgłoszeń. Algorytm nie wywala procesów, więc nowo przychodzący proces jest pierwszy w kolejce i czeka na zwolnienie procesora przez bieżąco wykonywany proces

-SJF (Shortest Job First) – najpierw najkrótsze zadanie - preferuje procesy, które mają najmniejsze wymagania odnośnie czasu procesora, potrzebnego na realizację przetwarzania.

- Zakleszczenia.

Zakleszczenie - sytuacja, gdy grupa procesów jest zatrzymana i czeka na zwolnienie jakiegoś zasobu. Przykładowo: proces A czeka na dostęp do drukarki zajętej przez proces B, a proces B czeka na dostęp do dysku, zajętego przez proces A. Dochodzi do zakleszczenia, ponieważ procesy “czekają na siebie wzajemnie”.

Gdyby ukazać zależności procesów typu “... czeka na ...” za pomocą grafu, to zakleszczeniem objęte są procesy zamknięte w cyklu:



Istnieją cztery **warunki konieczne**, które muszą być jednocześnie spełnione, aby system był podatny na zakleszczenia:

- wzajemne wykluczanie - procesy mogą się zatrzymywać w oczekiwaniu na zasoby zajęte w danym momencie przez inne procesy. W praktyce: albo z danego zasobu może jednocześnie korzystać maksymalnie **jeden** proces (np. drukarka obsługuje tylko jeden proces naraz), albo liczba procesów potencjalnie żądających dostępu przekracza liczbę egzemplarzy zasobu (np. do 10 procesów może żądać dostępu do dysku, ale ten akceptuje do 3 procesów jednocześnie; posiada tylko 3 egzemplarze zasobu "dysk")
- przetrzymywanie i oczekiwanie - proces **oczekuje** na zasób B, jednocześnie **przetrzymując** zasób A (i potencjalnie blokując dostęp do niego)
- brak wywłaszczenia - mechanizm ten (odbieranie przez system zasobów od procesu "siłą") nie występuje w systemie
- czekanie cykliczne - nie ma zapobiegania sytuacjom "wzajemnego czekania na siebie" (patrz rysunek powyżej)

Problem zakleszczeń można rozwiązać poprzez:

- zapobieganie (negując któryś z warunków koniecznych, np. implementując wywłaszczenie)
- unikanie (negując WSZYSTKIE warunki konieczne i ponadto monitorując przydziąły zasobów na bieżąco, np. sprawdzając algorytmem bankiera czy system jest w stanie bezpiecznym - czy w danym momencie nie ma zagrożenia zakleszczeniem)
- wykrywanie/usuwanie/odtwarzanie stanu procesów (algorytmy na: wykrywanie zakleszczeń, usuwanie przez wywłaszczenie, usuwanie przez przywrócenie stanu)

## **5. Zarządzanie pamięcią operacyjną LINUX oraz WINDOWS.**

Pamięć jest kluczowym zasobem systemu komputerowego dla wykonywania programów. Zarządzanie pamięcią jest jednak dość skomplikowane, gdyż jest ona jednocześnie wykorzystywana przez wiele procesów często różnych użytkowników oraz przez jądro systemu operacyjnego. Podstawowe zadania, realizowane w ramach zarządzania pamięcią obejmują przydział pamięci i jej odzyskiwanie, ochronę, udostępnianie w celu współdzielenia, transformację adresów oraz transfer danych pomiędzy poszczególnymi poziomami w hierarchii pamięci. Odpowiednie układy sprzętowe są odpowiedzialne za ochronę, transformację i w dużej części transfer danych, system operacyjny sprowadza się do dostarczenia odpowiednich danych tym układom

- **Pamięć wirtualna – organizacja pamięci - implementacja pamięci wirtualnej.**

Jest organizacją zasobów pamięci, zrealizowaną w oparciu o tzw. przestrzeń wymiany w pamięci drugiego rzędu. Daje możliwość powiększenia zasobów pamięci ponad dostępną pamięć fizyczną. Umożliwia bardziej racjonalne wykorzystanie pamięci operacyjnej, gdyż programy

tworzone są często z nadmiarem w stosunku do typowych potrzeb. Podstawą jej funkcjonowania jest mechanizm stronicowania.

- **Stronicowanie pamięci. LINUX oraz WINDOWS**

Jeden ze sposobów rozwiązywania problemu zewnętrznej fragmentacji, polegający na dopuszczeniu nieciągłości rozmieszczenia logicznej przestrzeni adresowej procesu w pamięci fizycznej. Jest to podział pamięci na mniejsze obszary o ustalonej lub zmiennej wielkości i przydzielanie tym blokom adresów fizycznych lub logicznych.

Podstawowa filozofia stronicowania:

- Pamięć fizyczna dzielona jest na bloki stałej długości zwane ramkami.
- Pamięć logiczna dzielona jest na bloki stałej długości zwane stronami.
- Rozmiary stron i ramek są identyczne.
- Przy wykonywaniu procesu, strony z pamięci pomocniczej wprowadzane są w odpowiednie ramki [pamięci operacyjnej](#).

- **Zadania mechanizmów stronicowania**

- odwzorowywanie adresów wirtualnych w adresy rzeczywiste, czyli:

\* określenie, do której strony odnosi się adres w programie

\* znalezienie ramy, którą aktualnie zajmuje dana strona.

- przesyłanie - w zależności od potrzeby - stron z pamięci pomocniczej do pamięci operacyjnej oraz odsyłanie już nieużywanych stron z powrotem

- **Segmentacja pamięci**

• Segmentacja jest schematem zarządzania pamięcią urzeczywistniającym sposób widzenia pamięci przez użytkownika.

• Przestrzeń adresów logicznych jest zbiorem segmentów.

- Segment to jednostka logiczna, taka jak: program główny, procedury, funkcje, metody, stosy, wykazy, tablice, zmienne itd.
- Dla ułatwienia implementacji segmenty są ponumerowane.

• Adres logiczny:<numer segmentu, odległość>

- Tablica segmentów: odwzorowuje dwuwymiarowe adresy logiczne w jednowymiarowe adresy fizyczne. Każda pozycja składa się z dwóch części:

- Baza segmentu – zawiera początkowy adres fizyczny segmentu w pamięci
- Granica segmentu - określa długość segmentu

Jedna z metod ochrony pamięci, używana przy wielozadaniowości. Każdy proces otrzymuje swój własny obszar pamięci, realizowany poprzez rejesty segmentowe. polega na podzieleniu przez procesor pamięci fizycznej na ciągłe bloki nazywane segmentami, które opisane są przez deskryptory zawierające:

- adres bazowy — fizyczny adres początku segmentu w pamięci
- rozmiar — długość segmentu w ustalonych jednostkach
- atrybuty określające rodzaj zawartości i dostępność (np. kod programu)
- identyfikator (określany też jako nazwa) — wartość wskazująca na opis segmentu w tablicy segmentów

- **Strategie przydziału pamięci.**

- Przydział ciągły - Polega na tym, że każdemu procesorowi przydziela się jeden spójny fragment pamięci. Pamięć jest podzielona na zajęte i wolne fragmenty, ponieważ aby system funkcjonował muszą działać jakieś procesy. Proces posiada adresy fizyczne od początku przydzielonego obszaru, do końca - 1, przez co nie wyjdzie on poza zakres swojej pamięci.
- Segmentacja - Podzielenie pamięci na poszczególne segmenty, np. kod, zmienne, stos, sterta, różni się od przydziału ciągłego tym, że program ma dostęp do kilku segmentów zamiast do jednego.
- Stronicowanie - Proces widzi spójny obcowaniem
- Połączenie stronicowania i segmentacji.

Proces widzi pamięć logiczną składającą się z szeregu segmentów. Poszczególny segmenty nie są rozmieszczone spójnie, a są szar pamięci logicznej, ale nie tworzy spójnego obszaru w pamięci fizycznej. Program widzi poszczególne strony, których wielkość jest określona potęgą dwójką np. 512b, 4kB itd. Program porusza się po pamięci za pomocą stron.

- Segmentacja ze stron podzielone na strony i każda strona może się znajdować w dowolnej ramce. Adres logiczny składa się z numeru segmentu i adresu w obrębie segmentu. Na podstawie numeru segmentu można wyznaczyć tablicę stron określającą rozmieszczenie stron tworzących segment.

- brak podziału, wolna przestrzeń adresowa w danej chwili przydzielana jednemu procesowi użytkowemu

- podział pamięci, wolna przestrzeń adresowa podzielona na części przydzielane pojedynczym procesom użytkowym
- wykorzystanie pamięci wirtualnej, istnieje jedna lub wiele wirtualnych przestrzeni adresowych przydzielanych procesom użytkowym, a mających w niewielkim stopniu pokrycie w pamięci operacyjnej.

- **Strategie wymiany dla systemów ze stronicowaniem.**

Proces żąda dostępu do strony w obszarze wymiany. W przypadku braku wolnych ramek

na dysk jest odsyłana strona:

4. Najdawniej używana:

- w deskryptorze (identyfikator pliku wykorzystywany przez system operacyjny) strony musi być zapisany czas odwołania do strony,
- stosowanie tego algorytmu jest obciążone kosztami zapisywania czasu odwołań do wszystkich stron.

5. Najmniej używana:

- w tym algorytmie należy odnotowywaćczęstość użycia strony,
- wadą tej metody jest fakt, że ostatnio załadowana strona może być nieroźważnie wymieniona,
- można zabronić wymiany stron załadowanych podczas określonego przedziału czasu.

6. Najdawniej załadowana:

- algorytm ten wymaga zapisu ciągu stron ładowanych do pamięci, co pozwala wskazać najdawniej załadowane strony,
- pomija się w nim fakt, że najwięcej odniesień może być do strony, która najdłużej znajduje się w pamięci.

## Algorytm FIFO zastępowania stron

Przyjrzyjmy się teraz, co się stanie, gdy liczba ramek wyniesie 4

| Chwila    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|
| Odwołanie | 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3  | 4  | 5  |
| Ramka 1   | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5  | 4  | 4  |
| Ramka 2   |   | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1  | 1  | 5  |
| Ramka 3   |   |   | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2  | 2  | 2  |
| Ramka 4   |   |   |   | 4 | 4 | 4 | 4 | 4 | 4 | 3  | 3  | 3  |

ofiara staje się strona, która najdłużej przebywa w pamięci

## Algorytm optymalny

| Chwila    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|
| Odwołanie | 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3  | 4  | 5  |
| Ramka 1   | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 4  | 4  |
| Ramka 2   |   | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2  | 2  | 2  |
| Ramka 3   |   |   | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  |
| Ramka 4   |   |   |   | 4 | 4 | 4 | 5 | 5 | 5 | 5  | 5  | 5  |

ofiaram staje się strona, która będzie nieużywana przez najdłuższy okres

## Algorytm LRU

| Chwila    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|
| Odwołanie | 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3  | 4  | 5  |
| Ramka 1   | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 5  |
| Ramka 2   |   | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2  | 2  | 2  |
| Ramka 3   |   |   | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 5  | 4  | 4  |
| Ramka 4   |   |   |   | 4 | 4 | 4 | 4 | 4 | 4 | 3  | 3  | 3  |

ofiaram strona, która nie była używana od najdłuższego czasu

## 6. System we/wy LINUX oraz WINDOWS

**Systemy wejścia-wyjścia** - mają na celu umożliwienie wprowadzania informacji do komputera (wpisywanie tekstu, odczytywanie i zapisywanie danych na urządzeniach pamięci masowej itp.) oraz jego wyprowadzanie (np. wyświetlanie na monitorze, drukowanie)

**Sprzęt wejścia-wyjścia:**

- pamięci masowe (dyski, taśmy)
- urządzenia komunikacji z użytkownikiem (monitory, klawiatury, myszki),
- urządzenie komunikacyjne (karty sieciowe, modemy)
- inne urządzenia specjalistyczne (kierownice, manetki, ...)

Urządzenia WE/WY pośredniczą w wymianie informacji między procesorem a układami zewnętrznymi. Korzystają z interfejsów, czyli ustalonych reguł i środków technicznych służących do połączenia komputera z urządzeniami zewnętrznymi. Interfejs składa się z :

- Wymagań logicznych
- Wymagań elektrycznych
- Wymagań konstrukcyjnych

Interfejs pozwala przesyłać dane równolegle oraz szeregowo w obu kierunkach jak i w jednym kierunku w zależności od potrzeb.

Dla procesora urządzenie WE/WY jest widoczne jako rejestr lub kilka rejestrów o określonych adresach i zestaw sygnałów sterujących.

## **7. Windows serwer – architektura oraz usługi sieciowe.**

**Windows Server** – nazwa grupy systemów operacyjnych firmy Microsoft przeznaczonych dla serwerów.

Do grupy tej należą:

- wersje podstawowe:
  - Windows Server 2003;
  - Windows Server 2008;
  - Windows Server 2012;
  - Windows Server 2016.
  - Windows Server 2019
- wersje specjalne:
  - Windows HPC Server 2008
  - Windows Small Business Server
  - Windows Essential Business Server
  - Windows Home Server

### **Usługi sieciowe:**

- **DNS**

Usługa DNS jest odpowiedzialna za przechowywanie i rozpoznawanie wszystkich nazw w sieci. Tłumaczy nazwę na adres IP. DNS (Domain Name Server) jest to hierarchiczny rozproszony system nazw sieciowych, który odpowiada na zapytania o nazwy domen.

- Różne rodzaje rekordów DNS:

- Rekord hosta (A lub AAAA).
- Rekord aliasu (CNAME).
- Rekord wymiany poczty (MX).
- Rekord serwera nazw (NS).

- **DHCP**

DHCP (Dynamic Host Configuration Protocol) – protokół dynamicznego konfigurowania węzłów. Jest to protokół komunikacyjny umożliwiający hostom uzyskanie od serwera danych konfiguracyjnych, takich jak:

- adres IP hosta,
- adres IP bramy sieciowej,
- adres serwera DNS,
- maska podsieci.

Brak serwera DHCP wymaga od administratora ręcznej konfiguracji wszystkich urządzeń w sieci.

- **Hyper-V**

W środowisku opartym na produktach firmy Microsoft stosuje się technologię zapewniającą możliwość uruchamiania maszyn wirtualnych (VM) - rola Hyper-V

Jest to oprogramowanie stosowane do wirtualizacji fizycznych maszyn, komputerów. Dzięki niemu można uruchamiać różne systemy operacyjne bez konieczności fizycznej ingerencji w już zainstalowany system operacyjny, na jednej fizycznej maszynie bez konieczności dzielenia dysku na partie. Wirtualny komputer ma własny, wirtualny sprzęt komputerowy i BIOS. Nazywa się go maszyną wirtualną.

Możliwości wirtualizacji zapewnione przez firmę Microsoft:

- Projektowanie i wdrażanie serwera Hyper-V.
- Korzystanie z wirtualnych przełączników.
- Implementacja serwera wirtualnego.
- Zarządzanie serwerem wirtualnym.
- Chronione maszyny wirtualne.
- Integracja z Linuksem.
- Serwer Hyper-V 2019.

- **Wirtualizacja serwerów**

Czym jest wirtualizacja serwerów? Najprościej rzecz ujmując jest to technologia, dzięki której aplikacje albo systemy uruchomione są na jednym serwerze i – co warto podkreślić – każde środowisko może działać w oparciu o inny system operacyjny. W obrębie serwera powstają więc mniejsze obszary wirtualne, które funkcjonują niezależnie od innych. Pozwala to na całkowite wykorzystanie zasobów serwera i jego mocy obliczeniowej, a w konsekwencji do szybszego funkcjonowania całego mechanizmu. To przekłada się zaś zarówno na optymalizację pracy wewnętrz firmy, jak i proces obsługi klientów. Zalety wynikające z wirtualizacji widać niemal gołym okiem. Nie ma chyba firmy działającej na rynku, której nie zależałooby na ograniczeniu ponoszonych kosztów. Coraz większy odsetek przedsiębiorców zauważa przy tym, że mniejsze zapotrzebowanie na dodatkowe sprzęty komputerowe, energię, czy po prostu przestrzeń, którą zajmie serwerownia jest w dłuższej perspektywie czasowej po prostu bardziej opłacalne. Oznacza to, że czas postawić na jakość, a zakup jednego, bardziej wydajnego serwera okazuje się po jakimś czasie znacznie lepszą inwestycją aniżeli kupowanie kilku słabszych i mniej wydajnych egzemplarzy. Choćby dlatego, że aby być na bieżąco i nadal świadczyć wysokiej jakości usługi trzeba mieć nowoczesny i bezawaryjny sprzęt... a żeby tak było należy go wymieniać co kilka lat.

## **8. Systemy operacyjne urządzeń mobilnych.**

- **Android** - system operacyjny z jądrem Linux dla urządzeń mobilnych. Język programowania: Java, z możliwością wstawek w C++. System otwarty. Dostępne źródła systemu. Oparty na Linuksie, z własną maszyną wirtualną Dalvik / ART. – Premiera systemu i pierwsze smartfony: połowa 2008
- **iOS** - Wywodzi się z systemu Mac Os X, do 2010 r. iPhone OS Oryginalnie powstał dla iPhone'a, później rozszerzony do obsługi iPoda, iPadu i AppleTV. Zamknięty system, brak możliwości instalacji aplikacji spoza AppStore . Język programowania: Objective-C
- **Windows10 Mobile** - Windows 10 Mobile to system operacyjny dla urządzeń mobilnych, który zastąpił Windows Phone.
- **Fuchsia** - Fuchsia jest systemem operacyjnym obecnie rozwijanym przez Google. Interfejs użytkownika i aplikacje Fuchsia są napisane z pomocą Flutter, frameworku, pozwalającego na międzyplatformowe możliwości deweloperskie dla systemów Fuchsia, Android i iOS.
- **Sailfish OS** - opracowany w Finlandii między innymi przez weteranów wywodzących się z Nokii (firmę Jolla). Jest dość popularny wśród developerów. Znajdziemy w tym systemie szereg zabezpieczeń oraz brak szpiegowskich usług od Google, pomimo wsparcia dla systemu Android.
- **Tizen** - również oparty na Linux system, mający długą historię, obecnie rozwijany przez firmę Samsung, która planuje jego wykorzystanie również w produkowanych przez siebie telewizorach. Jeśli Samsung zdecyduje się w pełni wspierać ten system na urządzeniach mobilnych (gdzie obecnie dominuje Android), na pewno zyska on istotną pozycję na rynku.
- **Ubuntu Touch** - system uniksopodobny będący mobilną wersją Ubuntu (jednej z dystrybucji systemu operacyjnego GNU/Linux)

## **9. Podstawowe właściwości systemów rozproszonych.**

Systemem rozproszonym nazywamy taki system, w którym przetwarzanie informacji odbywa się na wielu komputerach, często znacznie oddalonych geograficznie. Podstawowe właściwości:

-Dzielenie zasobów – wspólne użytkowanie obiektów, np. drukarki lub części pamięci

- Otwartość - zdolność do rozbudowy różnymi sposobami. System rozproszony może być rozszerzany o różne rozwiązania sprzętowe, np. dodawanie nowych urządzeń
- Skalowalność – możliwość zwiększania wybranych parametrów (przede wszystkim wydajności) w miarę potrzeb poprzez zwiększenie liczby elementów jest bardzo ważna właściwością
- Tolerowanie uszkodzeń – posiadanie nadmiarowości pewnych rozwiązań, które przy zadanym prawdopodobieństwie zapewniają działanie systemu mimo uszkodzenia
- Przeźroczystość - określana jako ukrywanie przed użytkownikiem części składowych systemu rozproszonego w taki sposób, że widzi go niezależnie od czasu i miejsca (bardziej) jako jedną całość niż zbiór elementów składowych
- Współbieżność-

## **10. Synchronizacja w systemach rozproszonych.**

Synchronizacja w systemach rozproszonych jest procesem bardziej złożonym niż w systemach scentralizowanych.

W zagadnieniach synchronizacji procesów na wielu maszynach nie jest istotna znajomość czasu absolutnego - istotne jest, aby procesy mogły ustalić kolejność zdarzeń.

Z tego powodu pomiar czasu możemy podzielić na:

- pomiar czasu fizyczny – zegary wskazują czas astronomiczny,
- pomiar czasu logiczny – zegary o wzajemnie uzgodnionym czasie, niekoniecznie astronomicznym.

## **11. Procesy w systemach rozproszonych.**

Ponieważ system rozproszony powinien być niewrażliwy na awarie poszczególnych maszyn, to w sytuacji, w której główny zarządcy systemu ulega awarii, na mamy do czynienia z blokadą całego systemu i oczekiwanie na decyzję zarządcy. Jest to sytuacja niedopuszczalna. Co więcej stworzenie procesu kontrolującego wszystko w systemie rozproszonym powoduje powstanie wąskiego gardła w postaci właśnie tego procesu. Musi on decydować o przydziałach wszystkich zasobów systemowych, przez co może zostać przeciążony w każdym systemie rozproszonym. Jego działanie wymagałoby zastosowanie specjalnych przywilejów i środków technicznych w celu realizacji jego roli (szybsza jednostka centralna, większa ilość pamięci operacyjnej, itp.) Poza tym o efektywności działania systemu rozproszonego decyduje właśnie grupa komputerów o podobnych parametrach, a nie jeden „superkomputer”.

## **Dla ubogich**

Proces rozproszony - powinien być niewrażliwy na awarie poszczególnych maszyn . Proces rozproszony jest definiowany jako pewien zbiór procesów sekwencyjnych, czyli takich które

reprezentują wykonanie pewnego ciągu operacji. Dodatkowo procesy mogą być współbieżne, a ich działanie jest skoordynowane, tak aby mogły zrealizować pewien wspólny cel.

Wyróżnia się dwa modele rozproszonych procesów:

- model statyczny (ang. static process model )
- model dynamiczny (ang. dynamic process model ).

W modelu statycznym zakłada się, że procesy, które tworzą pewne zadanie, tworzone są na początku jego działania. Drugi model nie nakłada takiego ograniczenia i procesy można tworzyć i niszczyć w dowolnym momencie trwania programu – zadania.

## **12. Bezpieczeństwo i ochrona LINUX oraz WINDOWS.**

**Linux** jest systemem rozprowadzanym na licencji GNU GPL (General Public License), co oznacza, że użytkownicy mogą kopiować, modyfikować i ponownie udostępniać kod źródłowy. Dzięki takiemu otwartemu podejściu programistom łatwiej jest znaleźć luki w systemie.

W systemie Linux za bezpieczeństwo odpowiada jądro systemu (kernel), wspomagane przez specjalistyczne oprogramowanie.

Podstawowe opcje ochrony w przypadku obydwu systemów są podobne. Są to: autentykacja, kontrola dostępu, logowanie zdarzeń, profile użytkowników i szyfrowanie.

The slide has a blue gradient background with a circular watermark logo in the center. The title 'Windows Defender ATP Exploit Guard' is at the top. Below it, the text 'Podstawowe elementy rozwiązania Defender ATP Exploit Guard:' is followed by a bulleted list of five items. At the bottom left, there is a small note 'autor: dr inż. Stanisława Plichta'.

**Windows Defender ATP Exploit Guard**

Podstawowe elementy rozwiązania Defender ATP Exploit Guard:

- Redukcja powierzchni ataku (*Attack Surface Reduction* - ASR).
- Ochrona sieci.
- Kontrolowany dostęp do folderu.
- Ochrona przed exploitami.

autor: dr inż. Stanisława Plichta

## Zapory systemu Windows

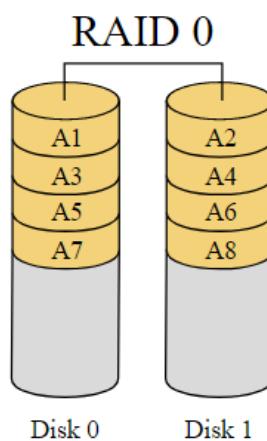
Istnieją trzy różne konsole, za pomocą których można konfigurować ustawienia zapory systemu Windows:

- Zapora Windows Defender (Panel sterowania),
- Zapora i ochrona sieci,
- Zapora Windows z zaawansowanymi zabezpieczeniami (WFAS)

### **13. Macierze dyskowe.**

#### RAID 0

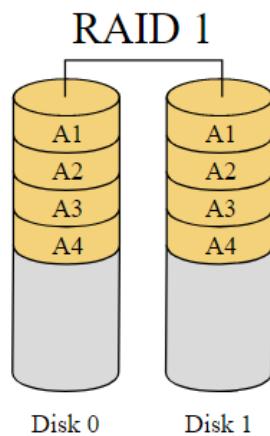
Jest to tzw. **stripe**. Dane zapisywane są na kilku dyskach, np. pierwszy blok na pierwszym dysku, drugi na drugim itd. Dzięki temu uzyskujemy znaczne przyspieszenie operacji odczytu i zapisu. Wadą tego systemu jest brak odporności na awarię. Jeśli dowolny dysk ulegnie uszkodzeniu, to najczęściej tracimy wszystkie dane, gdyż każdy dysk zawiera tylko część danych.



## RAID 1

Jest to tzw. **mirroring** (lustrzanie). Czyli połowa dysków zawiera dane a druga połowa jest ich dokładną kopią. W tym przypadku jesteśmy zabezpieczeni na awarię każdego z dysków, jednakże tracimy 50% całkowitej pojemności.

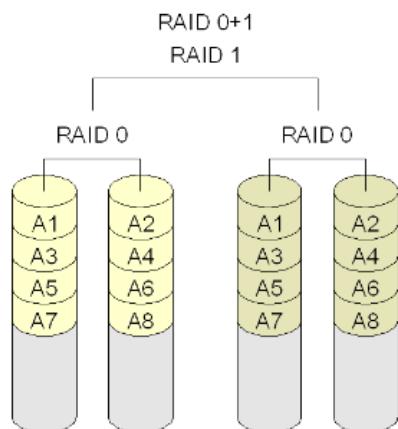
Przykład: dwa dyski, każdy o pojemności 1 TB połączone razem w RAID1, mają łączną pojemność 1 TB (drugi zawiera to samo co pierwszy).



## RAID 10

To połączenie RAID 0 z RAID 1, nazywany również RAID 1+0. W tym przypadku dane są zapisywane na połowie dysków jako **stripe**(RAID-0), a druga połowa jest ich lustrzaną kopią (RAID-1) – **mirroring**.

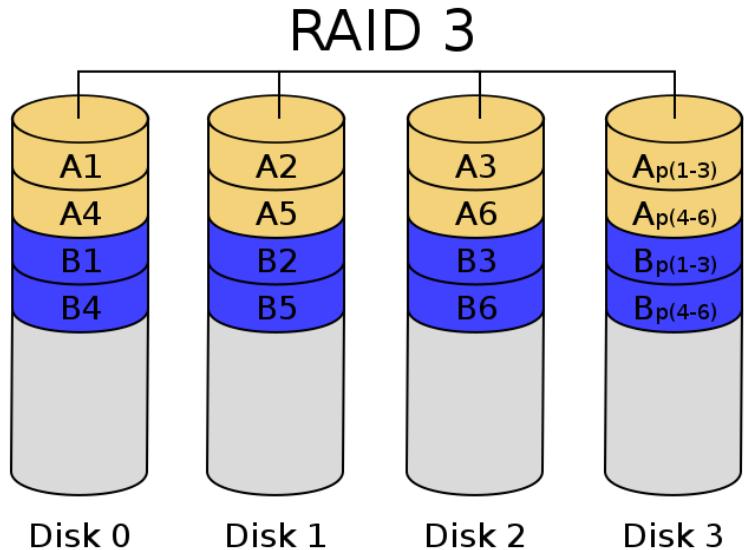
Jesteśmy odporni na awarię jednego dysku lub nawet dwóch, ale **nie dowolnych dwóch**. Należy jednak pamiętać, iż w przypadku awarii dysku i jego lustrzanej kopii (mirroru), **wszystkie dane zostaną utracone**. Z tego względu w macierzach NetApp serii FAS używany jest RAID-DP jako alternatywa dla RAID-10.



## RAID 3

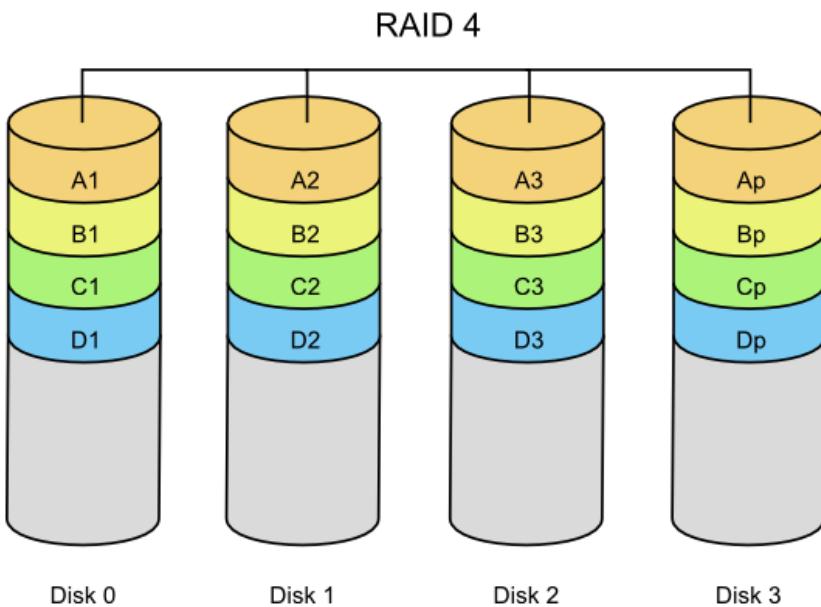
- Dane zapisywane na kilku dyskach - jeden dysk jest dyskiem parzystości.

- Wydajność macierzy jest gorsza niż w RAID 0 – wykorzystywana jest większa część przestrzeni dyskowej niż w RAID 1.
- Dyski są nierównomiernie obciążone - wada - dysk zapisujący informacje o parzystości jest obciążony znacznie bardziej niż pozostałe.



## RAID 4

W RAID 4 dane zapisywane są na kilku dyskach, a dodatkowo na **osobnym dysku są umieszcz sumy kontrolne**, które w przypadku awarii jednego z dysków umożliwiają jego odtworzenie poprzez wykonanie określonych operacji matematycznych.

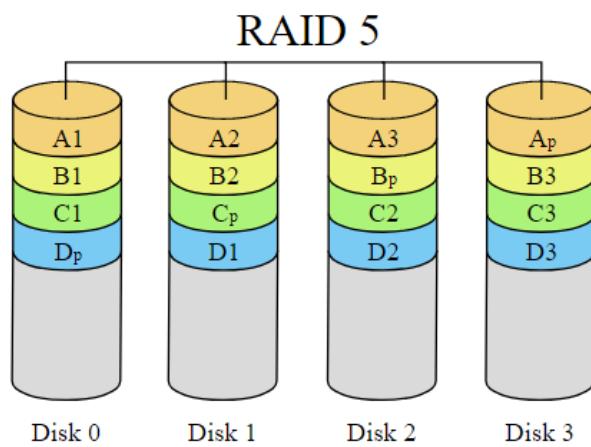


## RAID 5

W przypadku RAID-5 dane są również zapisane na wszystkich dostępnych dyskach, tak jak ma to miejsce w przypadku RAID4. Z tą różnicą jednak, że **częściowe sumy kontrolne są rozłożone na wszystkie dyski.**

Takie rozwiązanie ma jednak zasadniczą wadę. Po awarii jednego z dysków, macierz musi odbudować sumy kontrolne, obciążając mocno system. Drugi aspekt to rozbudowa – jeśli chcemy dołożyć do istniejącego systemu dodatkowy dysk, cała macierz musi zostać przebudowana.

RAID-5 jest odporny na awarię wyłączenie jednego dysku.



### **14. Standardy zabezpieczeń.**

Klasyfikowanie oprogramowania, włącznie z systemami operacyjnymi, według jasno określonych standardów pomaga rządowi, firmom i indywidualnym użytkownikom lepiej dbać o bezpieczeństwo poufnych i personalnych danych przechowywanych w systemach komputerowych.

| Klasa | Opis                                     |
|-------|--|
| A1    | Konstrukcja zweryfikowana                |
| B3    | Domeny bezpieczeństwa                    |
| B2    | Zabezpieczenie strukturalne              |
| B1    | Zabezpieczenie na bazie etykiet          |
| C2    | Kontrolowane zabezpieczenie dostępu      |
| C1    | Uznaniowa ochrona dostępu (przestarzałe) |
| D     | Minimalne zabezpieczenia                 |

## **Główne wymagania klasy C2**

- Istnienie mechanizmu bezpiecznego logowania.
- Uznaniowa kontrola dostępu.
- Monitorowanie zabezpieczeń.
- Zabezpieczenie przed ponownym użyciem obiektów. System Windows spełnia również dwa wymagania zabezpieczeń z poziomu B:
  - Funkcje zaufanej ścieżki.
  - Zarządzanie mechanizmami zaufanymi

## **Systemy**

- Systemy Windows 2000, Windows XP, Windows Server 2003 i Windows Vista otrzymały poświadczenie spełniania kryteriów CC zgodnie z profilem ochrony dostępu kontrolowanego.
- W czerwcu 2016 roku podobne certyfikaty uzyskały systemy Windows 10 i Windows Server 2012 R2

## **15. Zadanie z synchronizacji procesów.**

### **1.3. Egzamin A i B.**

#### **1. Wzajemne wyłączanie w systemach rozproszonych - omów algorytm scentralizowany.**

- Algorytm scentralizowany
- Algorytm rozproszony
- Algorytm pierścienia z żetonem (ang. Token ring)

#### **Algorytm scentralizowany**

W systemie istnieje jeden koordynator, tylko on może wyrazić zgodę na wejście procesu do sekcji krytycznej. Proces wysyła do koordynatora komunikat z **żądaniem**. Kiedy otrzyma od niego komunikat z **odpowiedzią**, wtedy może wejść do sekcji. Po wyjściu wysyła do koordynatora komunikat **zwalniający**.

#### **Algorytm rozproszony**

Proces, który chce wejść do sekcji krytycznej, buduje komunikat z nazwą tej sekcji, swoim numerem i bieżącym czasem.

- Wysyła komunikat do wszystkich innych procesów.
- Każdy komunikat jest potwierdzany.
- Jeśli proces otrzyma zamówienie od innego procesu, to działanie jakie podejmie, będzie zależeć od jego zamiarów związanych z sekcją krytyczną wymienioną w komunikacie.

### **Algorytm pierścienia z żetonem**

Gdy proces chce wejść do sekcji krytycznej:

- Oczekuje na żeton.
- Gdy otrzyma żeton wchodzi do sekcji.
- Po wyjściu przekazuje żeton do następnego procesu w pierścieniu.

### **2. Co to DNS i podaj rekordy.**

Usługa DNS jest odpowiedzialna za przechowywanie i rozpoznawanie wszystkich nazw w sieci. Tłumaczy nazwę na adres IP. DNS (Domain Name Server) jest to hierarchiczny rozproszony system nazw sieciowych, który odpowiada na zapytania o nazwy domen.

- Różne rodzaje rekordów DNS:
  - Rekord hosta (A lub AAAA).
  - Rekord aliasu (CNAME).
  - Rekord wymiany poczty (MX).
  - Rekord serwera nazw (NS).

### **3. Różnice link twardy a link symboliczny(UNIX)**

Linki umożliwiają nadanie wielu nazw jednemu plikowi.

Rozróżniamy dwa rodzaje linków:

- linki twarde
- linki symboliczne.

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.

- Link symboliczny jest plikiem, który wskazuje na nazwę innego pliku.
- Link twardy umożliwia tworzenie kilku nazw dla jednego i- węzła.

#### **4. Mechanizm planowania (WINDOWS). Gdzie implementowany jest kod.**

System Windows implementuje mechanizm wywłaszczeniowego planowania sterowanego za pomocą priorytetów.

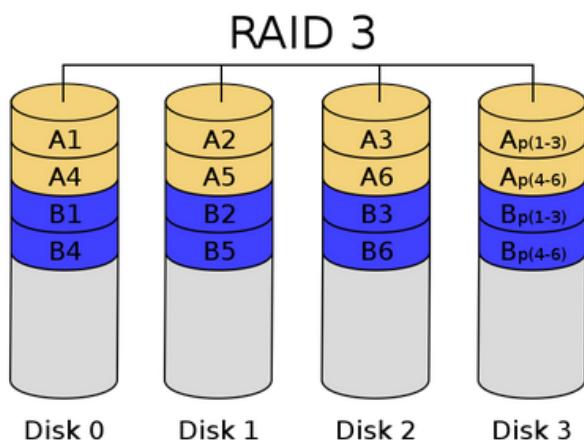
- Po wybraniu wątku do uruchomienia działa on przez ilość czasu nazywaną kwantem.
- Wartości kwantu mogą się zmieniać dla poszczególnych systemów i procesów z trzech następujących powodów:
  - Ustawienia konfiguracji systemu
  - Pierwszoplanowy lub drugoplanowy status procesu.
  - Użycie obiektu zadania do zmiany kwantu czasu.

Kod planowania w systemie Windows implementowany jest w jądrze - nie istnieje żaden moduł planowania ani funkcja.

#### **5. Opisać Macierze co to. Raid 3.**

##### **RAID 3**

- Dane zapisywane na kilku dyskach - jeden dysk jest dyskiem parzystości.
- Wydajność macierzy jest gorsza niż w RAID 0 – wykorzystywana jest większa część przestrzeni dyskowej niż w RAID 1.
- Dyski są nierównomiernie obciążone - wada - dysk zapisujący informacje o parzystości jest obciążony znacznie bardziej niż pozostałe.



<https://poradnik.kz1.pl/index.php?page=macierzeraid>

#### **6. Mobilne systemy operacyjne.**

- **Android** - system operacyjny z jądrem Linux dla urządzeń mobilnych. Język programowania: Java, z możliwością wstawek w C++. System otwarty. Dostępne źródła systemu. Oparty na Linuksie, z własną maszyną wirtualną Dalvik / ART. – Premiera systemu i pierwsze smartfony: połowa 2008

- **iOS** - Wywodzi się z systemu Mac Os X, do 2010 r. iPhone OS Oryginalnie powstał dla iPhone'a, później rozszerzony do obsługi iPoda, iPadai AppleTV. Zamknięty system, brak możliwości instalacji aplikacji spoza AppStore . Język programowania: Objective-C
- **Windows10 Mobile** - Windows 10 Mobile to system operacyjny dla urządzeń mobilnych, który zastąpił Windows Phone.
- **Fuchsia** - Fuchsia jest systemem operacyjnym obecnie rozwijanym przez Google. Interfejs użytkownika i aplikacje Fuchsia są napisane z pomocą Flutter, frameworku, pozwalającego na międzyplatformowe możliwości deweloperskie dla systemów Fuchsia, Android i iOS.
- **Sailfish OS** - opracowany w Finlandii między innymi przez weteranów wywodzących się z Nokii (firmę Jolla). Jest dość popularny wśród developerów. Znajdziemy w tym systemie szereg zabezpieczeń oraz brak szpiegowskich usług od Google, pomimo wsparcia dla systemu Android.
- **Tizen** - również oparty na Linux system, mający długą historię, obecnie rozwijany przez firmę Samsung, która planuje jego wykorzystanie również w produkowanych przez siebie telewizorach. Jeśli Samsung zdecyduje się w pełni wspierać ten system na urządzeniach mobilnych (gdzie obecnie dominuje Android), na pewno zyska on istotną pozycję na rynku.
- **Ubuntu Touch** - system uniksopodobny będący mobilną wersją Ubuntu (jednej z dystrybucji systemu operacyjnego GNU/Linux)

## 7. Co to Sekcja krytyczna. Sposoby imlementacji.

**Sekcja krytyczna** – fragment kodu programu, w którym korzysta się z **zasobu dzielnego**, a co za tym idzie w danej chwili może być wykorzystywany przez co najwyżej jeden wątek.

## 8. Wyjaśnić stronicowanie i odwzorowywanie adresów.

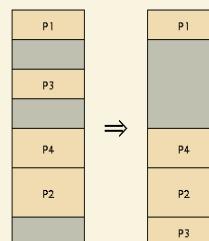
Stronicowanie to rozwiązanie, w którym proces widzi spójny obszar pamięci logicznej, ale nie tworzy ona spójnego obszaru w pamięci fizycznej. Zarówno pamięć logiczna, jak i pamięć fizyczna jest podzielona na kawałki **równiej** wielkości. W odniesieniu do pamięci logicznej mówimy o stronach, a w odniesieniu do pamięci fizycznej mówimy o ramkach. Wielkość stron i ramek jest potęgą dwójki z przedziału od 512B do 16MB. Typowa wielkość to 4kB. Strony i ramki są podstawowymi jednostkami przydziału pamięci. W rezultacie, wielkość przydzielonego procesowi obszaru jest wielokrotnością wielkości ramek i stron. Strony pamięci logicznej mogą być umieszczone w dowolnych ramkach pamięci fizycznej. Nie muszą tworzyć spójnego obszaru w pamięci fizycznej, ani nie muszą występować w określonej kolejności.

## 9. Fragmentacja zewnętrzna i wewnętrzna.

### Fragmentacja zewnętrzna i kompaktyfikacja

W momencie uruchomienia procesu, system operacyjny musi przydzielić procesowi pamięć. Wolna pamięć fizyczna może być poszatkowana przydzielonymi już obszarami na wiele fragmentów. Może się więc tak zdarzyć, że łączna ilość wolnej pamięci będzie wystarczająca, ale nie będzie jednego spójnego obszaru, wystarczająco dużego, żeby przydzielić z niego pamięć dla procesu. Zjawisko takie nazywamy fragmentacją zewnętrzną.

Jeżeli mamy do czynienia z fragmentacją zewnętrzną, to konieczne staje się scalenie wolnych obszarów. Scalanie takie nazywamy kompaktyfikacją lub upakowaniem. Kompaktyfikacja jest kosztowna, ze względu na konieczność kopирования dużych obszarów pamięci fizycznej. Dlatego też należy starać się unikać fragmentacji zewnętrznej. Poniższy rysunek przedstawia przykładową kompaktyfikację. Zwróćmy uwagę na to, że zajęte obszary nie muszą być upakowane razem, scalamy obszary wolne.



### Fragmentacja wewnętrzna

Stosując stronicowanie unikamy fragmentacji zewnętrznej. Bez względu na to jak są rozsiane po pamięci fizycznej wolne ramki, możemy z nich utworzyć spójny obszar pamięci logicznej. Nadal nie jesteśmy jednak w stanie wykorzystać całej pamięci. Jednostką przydziału pamięci jest strona/ramka. Zwykle więc część ostatniej strony pozostanie niewykorzystana. Z pewnym uproszczeniem, możemy przyjąć, że średnio połowa ostatniej strony w przydzielanym obszarze jest niewykorzystana. Zjawisko to nazywamy **fragmentacją wewnętrzna**. Fragmentacja wewnętrzna występuje wszędzie tam, gdzie wielkość przydzielanej pamięci musi być wielokrotnością pewnej większej jednostki, np. strony. Fragmentację wewnętrzna można zmniejszyć zmniejszając wielkość strony. To jednak oznacza zwiększenie rozmiaru tablicy stron. Wybierając wielkość stron należy wziąć pod uwagę zarówno straty związane z fragmentacją wewnętrzna, jak i straty związane z przechowywaniem tablicy stron.

### Definicja zewnętrznej fragmentacji

Fragmentacja zewnętrzna występuje wtedy, gdy w pamięci znajduje się wystarczająca ilość miejsca, aby spełnić żądanie pamięciowe procesu. Ale żądanie pamięci procesu nie może zostać spełnione, ponieważ dostępna pamięć jest w sposób nieciągły. Albo zastosujesz strategię alokacji pamięci pierwszego dopasowania, albo najlepszego dopasowania, która spowoduje zewnętrzna fragmentację.

### Definicja wewnętrznej fragmentacji

Wewnętrzna fragmentacja występuje, gdy pamięć jest podzielona na **bloki o ustalonej wielkości**. Ilekroć żądanie procesu dla pamięci, blok o stałej wielkości jest przydzielany do procesu. W przypadku, gdy pamięć przypisana do procesu jest nieco większa niż wymagana pamięć, różnica między pamięcią przypisaną a żądaną jest **wewnętrczną fragmentacją**.

## 10. Planisci.

- Planista średnioterminowy odpowiedzialny jest za wymianę (swapping) procesów między pamięcią operacyjną a dyskiem – stosowany często w systemach z podziałem czasu.
- Zadanie planisty długoterminowego - dobranie dobrej mieszanki procesów (process mix) zawierającej procesy obydwu rodzajów
- Planista krótko terminowy odpowiedzialny jest za przejście ze stanu gotowości do wykonania

### Co za chuj to usunął

- Krótkoterminowy – planista przydziału procesora, który podejmuje decyzję, kiedy proces ma przejść ze stanu gotowego w wykonywanym
- Średnioterminowy – odpowiada za decyzję, które procesy usunąć z pamięci, a które ponownie załadować.
- Długoterminowy – planista zadań, podejmuje decyzję o przyjęciu nowego procesu do systemu i kontrolą zadań w systemie oraz ich doborem do wykorzystania zasobów

## 11. Wymiany stron.

### Algorytm FIFO zastępowania stron

Przyjrzyjmy się teraz, co się stanie, gdy liczba ramek wyniesie 4

| Chwila    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|
| Odwołanie | 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3  | 4  | 5  |
| Ramka 1   | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5  | 4  | 4  |
| Ramka 2   |   | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1  | 1  | 5  |
| Ramka 3   |   |   | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2  | 2  | 2  |
| Ramka 4   |   |   |   | 4 | 4 | 4 | 4 | 4 | 4 | 3  | 3  | 3  |

ofiara staje się strona, która najdłużej przebywa w pamięci

### Algorytm optymalny

| Chwila    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|
| Odwołanie | 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3  | 4  | 5  |
| Ramka 1   | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 4  | 4  |
| Ramka 2   |   | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2  | 2  | 2  |
| Ramka 3   |   |   | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  |
| Ramka 4   |   |   |   | 4 | 4 | 4 | 5 | 5 | 5 | 5  | 5  | 5  |

ofiara staje się strona, która będzie nieużywana przez najdłuższy okres

## Algorytm LRU

| Chwila    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------|---|---|---|---|---|---|---|---|---|----|----|----|
| Odwółanie | 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3  | 4  | 5  |
| Ramka 1   | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 5  |
| Ramka 2   |   | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2  | 2  | 2  |
| Ramka 3   |   |   | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 5  | 4  | 4  |
| Ramka 4   |   |   |   | 4 | 4 | 4 | 4 | 4 | 4 | 3  | 3  | 3  |

ofiara strona, która nie była używana od najdłuższego czasu

### 12. Stany wątków w windows(10??)

**Wątek** (ang. *thread*) – część programu wykonywana współbieżnie w obrębie jednego procesu; w jednym procesie może istnieć wiele wątków.

- Realizacja wątków na poziomie jądra systemu operacyjnego — jądro tworzy odpowiednie struktury (blok kontrolny) do utrzymywania stanu wątku.
- Realizacja wątków na poziomie użytkownika — struktury związane ze stanami wątków tworzone są w przestrzeni adresowej procesu.

Systemy operacyjne



### Stany wątku w systemie Windows 2000/XP

- Inicjalizowany (initialized, wartość 0) — stan wewnętrzny w trakcie tworzenia wątku,
- Gotowy (ready, wartość 1) — oczekuje na przydział procesora,
- Wykonywany (running, wartość 2),
- Czuwający (standby, wartość 3) — wybrany do wykonania jako następny,
- Zakończony (terminated, wartość 4),
- Oczekujący (waiting, wart. 5) — oczekuje na zdarzenie,
- Przejście (transition, wartość 6) — oczekuje na sprowadzenie swojego stosu jądra z pliku wymiany,
- Unknown (wart. 7)

### **13. Struktura domenowa.**

Co to kurwa jest ja opisywalem na egzaminie klient-serwer i jakims cudem punkta mi dała

### **14. Komponenty centrum wykonawczego Windows.**

#### **Centrum wykonawcze**

- Centrum wykonawcze systemu Windows jest górną warstwą biblioteki *Ntoskrnl.exe* (dolną jest jądro).
- Zawiera następujące rodzaje funkcji:
  - Funkcje eksportowane i wywoływalne z trybu użytkownika – biblioteki *Ntdll.dll*
  - Funkcje sterowników urządzeń wywoływanie za pomocą funkcji *DeviceControl*,
  - Funkcje eksportowane, które można wywoływać tylko z poziomu jądra,
  - Funkcje, które są zdefiniowane jako symbole globalne, ale nie są eksportowane,
  - Funkcje wewnętrzne, ale nie będące symbolami globalnymi.

#### **Centrum wykonawcze - komponenty**

Centrum wykonawcze zawiera następujące komponenty:

- Menedżer konfiguracji,
- Menedżer procesów,
- Monitor bezpieczeństwa odwołań (SRM),
- Menedżer wejścia-wyjścia,
- Menedżer urządzeń Plug-and-Play,
- Menedżer zasilania,
- Menedżer pamięci,
- Menedżer obiektów,
- ALPC.

### **15. Zadania i komponenty menedżera pamięci WINDOWS.**

## Zadania Menedżera pamięci

- Odwzorować (zmapować) wirtualną przestrzeń adresową procesu na pamięć fizyczną, aby w momencie, gdy wątek działający w kontekście tego procesu będzie odczytywał dane lub je zapisywał do wirtualnej przestrzeni adresowej, nastąpiło odwołanie do odpowiedniego adresu fizycznego.
- Przeprowadzić stronicowanie części zawartości pamięci na dysk w momencie, gdy działające wątki starają się użyć więcej pamięci fizycznej, niż jest aktualnie dostępne, i ponownie przenieść tę zawartość do pamięci fizycznej, gdy jest potrzebna.

## Komponenty Menedżera pamięci

- Zestaw systemowych usług wykonawczych do przydzielania i zwalniania pamięci oraz zarządzania tą pamięcią.
- Uchwyt nieważnej operacji tłumaczenia i pułapki błędu dostępu potrzebny do obsługi wykrytych sprzętowo wyjątków w zarządzaniu pamięcią oraz przekształcania stron wirtualnych na rezydentne w imieniu procesu.
- Sześć kluczowych procedur, z których każda działa w jednym z sześciu różnych wątków trybu jądra w ramach procesu *System*.
  - Menedżer zestawu równowagi
  - Procedura wymiany procesu/stosu
  - Moduł zapisu stron zmodyfikowanych
  - Moduł zapisu stron zmapowanych
  - Wątek dereferencji segmentu
  - Wątek zerowania stron

### 1.4. Egzamin C i D.

1. Planowanie w systemie Windows.

2. Stany wątków w Windows(jest wyżej).

3.Zamiana adresów wirtualnych na fizyczne.

4. Wyjaśnij pojęcie szamotania.

Mianem **szamotania** określany jest stan procesu, w którym spędza on więcej czasu na oczekiwaniu na brakujące strony pamięci niż na faktycznym wykonywaniu obliczeń, co znacząco spowalnia jego działanie. Problem szamotania występuje we wszystkich implementacjach i objawia się przy zbyt dużym zapotrzebowaniu na pamięć ze strony procesów przy zbyt małej liczbie wolnych ramek. Problem może być częściowo rozwiązany przez poprawienie jakości programów, lecz na dłuższą metę jedynym skutecznym sposobem jest zainstalowanie większej ilości fizycznej pamięci operacyjnej RAM.

## **5. DHCP.**

- **DHCP**

DHCP ([Dynamic Host Configuration Protocol](#)) – protokół dynamicznego konfigurowania węzłów. Jest to protokół komunikacyjny umożliwiający hostom uzyskanie od serwera danych konfiguracyjnych, takich jak:

- adres IP hosta,
- adres IP bramy sieciowej,
- adres serwera DNS,
- maska podsieci.

Brak serwera DHCP wymaga od administratora ręcznej konfiguracji wszystkich urządzeń w sieci.

## **6. Systemy mobilne (wyżej).**

## **7. Zakleszczenia.**

# Zakleszczenia

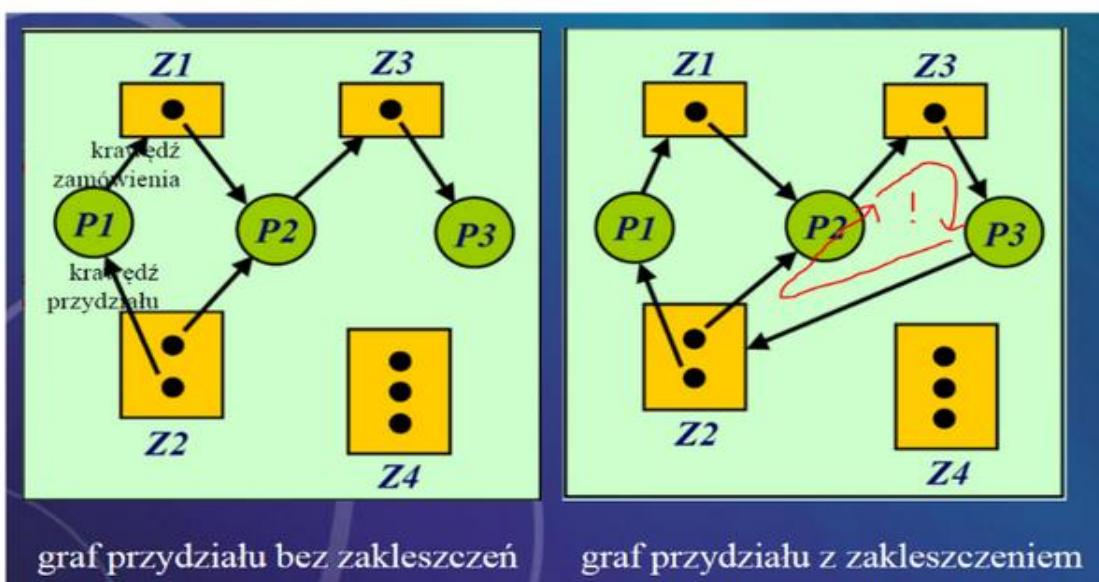
Zbiór procesów będących w impasie wywołanym przez to, że każdy proces należący do tego zbioru przetrzymuje zasoby potrzebne innym procesom z tego zbioru, a jednocześnie czeka na zasoby przydzielone innym procesom.

- Zakleszczenie jest zjawiskiem niezwykle niepożądany w systemie i dlatego opracowano rozmaite metody radzenia sobie z nimi.

- Zakleszczenia.

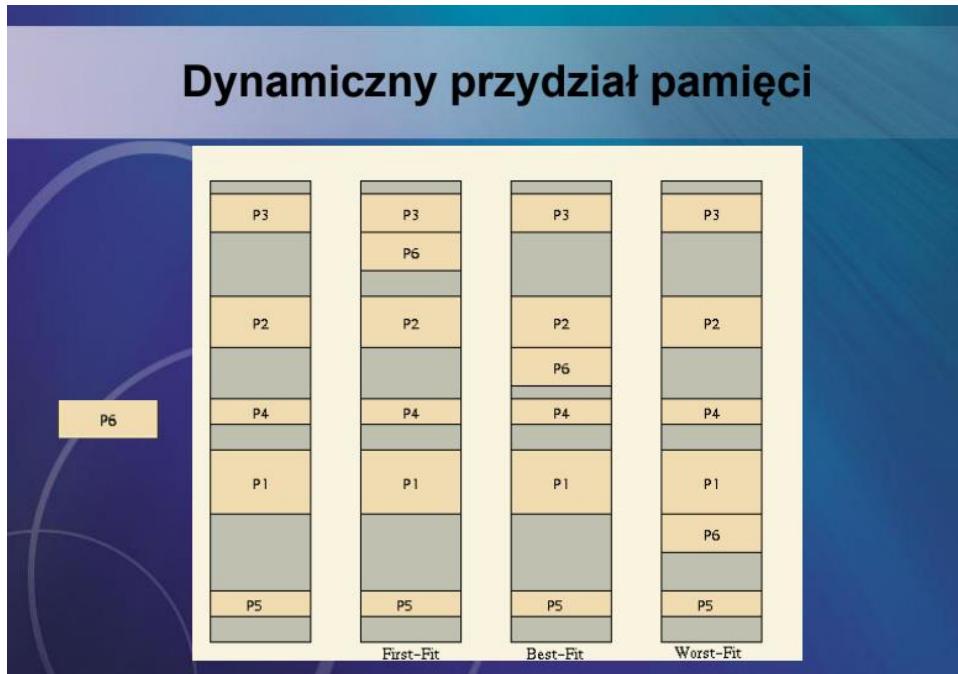
Zakleszczenie - sytuacja, gdy grupa procesów jest zatrzymana i czeka na zwolnienie jakiegoś zasobu. Przykładowo: proces A czeka na dostęp do drukarki zajętej przez proces B, a proces B czeka na dostęp do dysku, zajętego przez proces A. Dochodzi do zakleszczenia, ponieważ procesy "czekają na siebie wzajemnie".

Gdyby ukazać zależności procesów typu "... czeka na ..." za pomocą grafu, to zakleszczeniem objęte są procesy zamknięte w cyklu:



## 8. Obsługa wejścia/wyjścia.

## 9. Dynamiczne przydzielane pamięci.



Sposoby:

-First fit

Wybierany jest pierwszy wolny obszar, który ma wystarczającą ilość miejsca

-Best fit

Wybierany jest najmniejszy wolny obszar, który ma wystarczającą ilość miejsca

-Worst fit

Wybierany jest największy wolny obszar, który ma wystarczającą ilość miejsca

## 10. Systemy smp asmp.

**Wieloprocesorowość symetryczna, przetwarzanie symetryczne, SMP** (ang. *symmetric multiprocessing*) – architektura komputerowa, która pozwala na znaczne zwiększenie mocy obliczeniowej systemu komputerowego poprzez wykorzystanie dwóch lub większej liczby procesorów do jednoczesnego wykonywania zadań. Szczególnym przypadkiem zastosowania tej architektury jest procesor wielordzeniowy, który można traktować jako system wieloprocesorowy z procesorami jednordzeniowymi.

W architekturze SMP procesory współdzielą zasoby pamięci oraz urządzenia wejścia/wyjścia przy pomocy magistrali systemowej, a każdy z tych procesorów może zostać przypisany do wykonywania

konkretnego zadania, tak aby w systemie następowało równoważenie obciążenia (ang. *load balancing*)

Ważną rzeczą przy przetwarzaniu symetrycznym jest stosowanie wielowątkowości przez uruchomione programy komputerowe i, co za tym idzie, jej obsługa przez system operacyjny – upraszcza to w znacznym stopniu możliwość podziału poszczególnych procesów, tak aby mogły być przetwarzane przez kilka procesorów jednocześnie. Systemami operacyjnymi obsługującymi architekturę SMP jest większość Uniksów, Windows NT oraz BeOS

**Wieloprocesorowość asymetryczna** (ang. *ASymmetric MultiProcessing, ASMP*) – architektura komputerowa mająca na celu zwiększenie mocy obliczeniowej i funkcjonalności komputera poprzez wykorzystanie kilku procesorów.

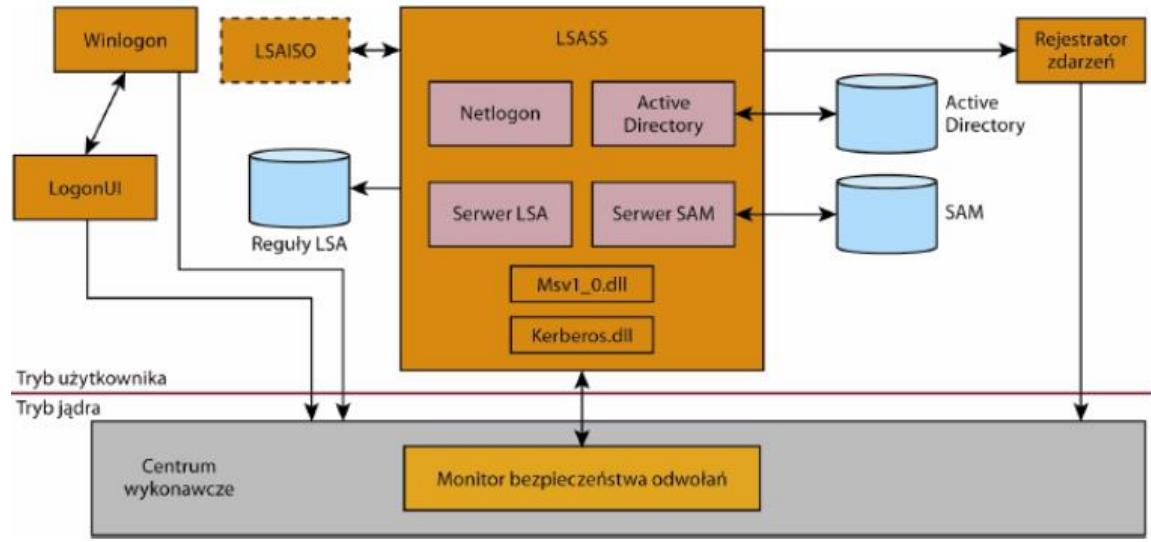
W architekturze asymetrycznej wszystkie procesory nie są traktowane jednakowo, jak w architekturze symetrycznej, lecz niektórym z nich przydzielane są zadania specjalne. Jeden z procesorów może np. obsługiwać wyłącznie operacje wejścia/wyjścia, podczas gdy pozostałe zajmują się obsługą typowych aplikacji. Procesory w układzie ASMP mogą mieć dostęp do rozłącznych zestawów urządzeń peryferyjnych. Takie rozwiązanie upraszcza projektowanie systemów, ale ogranicza maksymalną wydajność w porównaniu do architektury symetrycznej (SMP).

## 11. Fragmentacja zewn/wewnt.

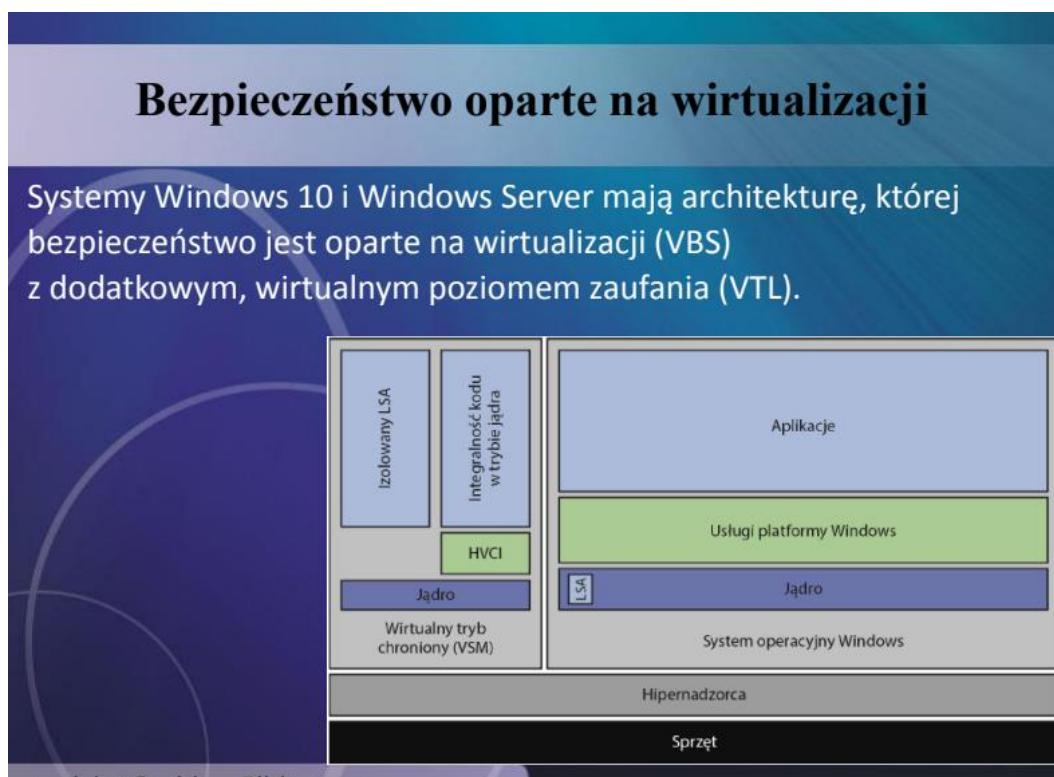
Wyżej test A i B

## 12. Bezpieczeństwo w windows komponenty.

### Komponenty odpowiedzialne za bezpieczeństwo systemu Windows



## Centrum zabezpieczeń systemu windows



### 13. NTFS.

Wyżej w zagadnieniach

### 14. Algorytm rozproszony.

#### Algorytm rozproszony

Proces, który chce wejść do sekcji krytycznej, buduje komunikat z nazwą tej sekcji, swoim numerem i bieżącym czasem.

- Wysyła komunikat do wszystkich innych procesów.
- Każdy komunikat jest potwierdzany.
- Jeśli proces otrzyma zamówienie od innego procesu, to działanie jakie podejmie, będzie zależeć od jego zamiarów związanych z sekcją krytyczną wymienioną w komunikacie.

### 15. ????????

# Z KOLOSÓW WSTAWIĆ GDZIEŚ

XXXXXXXXXXXXXXXXXXXXXX

Omów metody zarządzania pamięcią

(4/4 pkt)

Aby pamięć była dobrze zarządzana potrzebuje ona odpowiedniej strategii przydzielu pamięci. System operacyjny musi wiedzieć z którego obszaru przydzielić pamięć. Obszar ten musi być wystarczająco duży.

Sposoby:

-First fit

Wybierany jest pierwszy wolny obszar, który ma wystarczającą ilość miejsca

-Best fit

Wybierany jest najmniejszy wolny obszar, który ma wystarczającą ilość miejsca

-Worst fit

Wybierany jest największy wolny obszar, który ma wystarczającą ilość miejsca

Strategie:

Przydział ciągły - Polega na tym, że każdemu procesorowi przydziela się jeden spójny fragment pamięci. Pamięć jest podzielona na zajęte i wolne fragmenty, ponieważ aby system funkcjonował muszą działać jakieś procesy. Proces posiada adresy fizyczne od początku przydzielonego obszaru, do końca - 1, przez co nie wyjdzie on poza zakres swojej pamięci.

Segmentacja - Podzielenie pamięci na poszczególne segmenty, np. kod, zmienne, stos, sterta, różni się od przydziału

ciągłego tym, że program ma dostęp do kilku segmentów zamiast do jednego.

Stronicowanie - Proces widzi spójny obszar pamięci logicznej, ale nie tworzy spójnego obszaru w pamięci fizycznej.

Program widzi poszczególne strony, których wielkość jest określona potęgą dwójką np. 512b, 4kB itd.  
Program

poruszają się po pamięci za pomocą stron.

Segmentacja ze stronicowaniem - Połączenie stronicowania i segmentacji. Proces widzi pamięć logiczną składającą

się z szeregu segmentów. Poszczególny segmenty nie są rozmieszczone spójnie, a są podzielone na strony i każda

strona może się znajdować w dowolnej ramce. Adres logiczny składa się z numeru segmentu i adresu w obrębie

segmentu. Na podstawie numeru segmentu można wyznaczyć tablicę stron określającą rozmieszczenie stron

tworzących segment

XXXXXXXXXXXXXXXXXXXXXX

Na czym polega strategia zbioru roboczego i do czego służy

[3/3 pkt)

Zbiór roboczy procesu jest zbiorem stron zaadresowanych w ciągu ostatnich D\* odniesień do pamięci. Mając ustalone wielkości zbiorów roboczych można przydzielić procesom taką ilość pamięci, która jest odpowiednia dla ich zbiorów roboczych.

Strategia zbioru roboczego służy do zapobiegania szamotaniu (spędzaniu większej ilości czasu na czekaniu na brakujące strony niż na wykonywaniu obliczeń) utrzymując stopień wieloprogramowości na wysokim poziomie. Zaletą jest optymalizacja użycia procesora.

D - Stała którą należy dobrać w zależności od programu, musi zawierać przewidywaną liczbę instrukcji. W przypadku złego doboru wartości np. zbyt małej nie zostanie uchwycona cała strefa, a w przypadku zbyt dużej połączą się rozmiary kilku stref.

XXXXXXXXXXXXXXXXXXXXXX

wyjaśnij pojęcie anomalii Belady'ego

(1/1 punkt)

Anomalia Belady'ego polega na zwiększaniu się ilości błędów mimo (intuicyjnie lepszego) zwiększenia ilości ramek.

Jeśli dla jakiejś liczby ramek pojawi się jakaś liczba błędów, to dla większej liczby ramek może wbrew pozorom

pojawić się większa ilość błędów. Skutkuje to możliwością spowolnienia algorytmu w sytuacji, gdy zwiększymy

dostępna pamięć.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

4. Opisz tworzenie procesu. Opisz co dzieje się z procesem po jego utworzeniu

(4/4 pkt)

Użytkownik zleca uruchomienie programu, system wypełnia strukturę opisującą proces, system kopiuje przestrzeń adresową, dane i kod z pliku wykonywalnego mapuje zasoby systemowe w przestrzeni adresowej procesu, ustawia stan procesu na działający, dodaje proces do kolejki procesów oczekujących na procesor, zwraca sterowanie powłoce systemowej.

najpierw utworzenie procesu potomnego fork()

W procesie macierzystym funkcja zwraca PID procesu potomnego, zwraca identyfikator procesu getpid()

[identyfikator procesu który ją wywołał], proces zaczyna wykonywanie programu którego kod znajduje się w path i

zwraca PID procesu który się zakończy (0 ile się zakończy!, bo może jeszcze trafić w stan czekania, gdzie czeka na

np przerwanie lub jest w stanie gotowości i czeka na ponowne przydzielanie procesora, zombie)

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

2. Omów wywieszające algorytmy szeregowania zadań [q]

(2/2 pkt)

W przypadku planowania priorytetowego, gdy jakiś proces ma wyższy priorytet od aktualnie wykonywanego procesu, wykonywany proces o niższym priorytecie jest natychmiastowo wywieszczany i zawieszany, a procesor zostaje przyznany procesowi o wyższym priorytecie. W planowaniu rotacyjnym procesy są ustawione w cyklicznej kolejce, gdzie po kolei system przydziela im procesor

XXXXXXXXXXXXXXXXXXXXXX

## 5. Przybliż sposoby komunikowania się między procesami w systemie Unix

(3/3 pkt)

W celu komunikacji między procesami w systemie UNIX wykorzystuje się:

- przekazywanie komunikatów
- pamięć dzielona - najszybsza metoda komunikacji między procesowej
- łącza komunikacyjne - Najprostszy sposób przekazywania danych między procesami. Są 2 rodzaje łączy: z kolejką FIFO i nienazwane

- łącza nazwane (FIFO) - służą do komunikacji na zasadzie kolejki prostej. Kolejka FIFO umożliwia wymianę informacji pomiędzy procesami nieznanymi ze sobą

-Sygnały

-Pliki

-Semafora