Wykład 6

80486
Mechanizmy ochrony zadań

Bartłomiej Zieliński, PhD, DSc

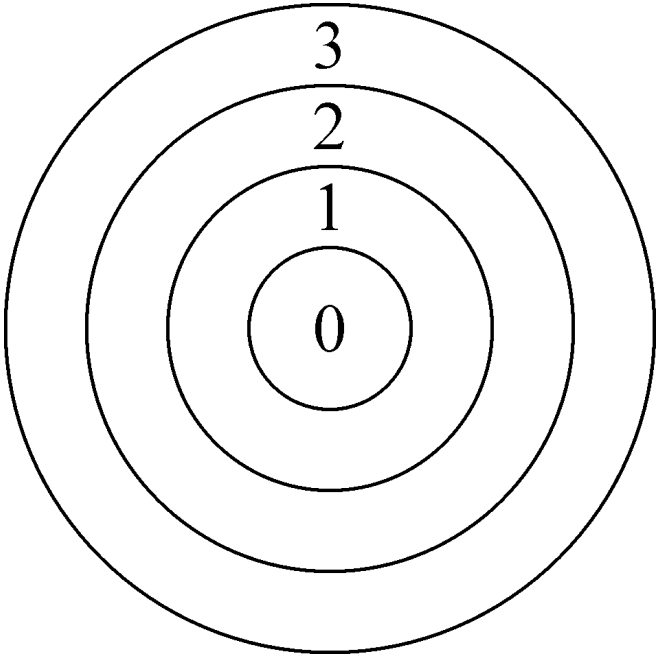
486 – ochrona zadań

Program:

- Model ochrony zadań
- Segment stanu zadania
- Struktura i zastosowania deskryptorów bramek
- Przewania i wyjątki
- Pamięć podręczna, bufory zapisu, przesyły seryjne

486 – ochrona zadań

- Poziomy uprzywilejowania
  - 0: jądro syst.oper.
  - 1: funkcje syst.oper.
  - 2: prog. pośredniczące
  - 3: programy użytkownika
- RPL, CPL, DPL
  - CPL może się zmienić
    - (np. wywołanie funkcji systemowej)
  - Dostęp do segmentów danych
    - DPL ≥ CPL & DPL ≥ RPL



486 – ochrona zadań

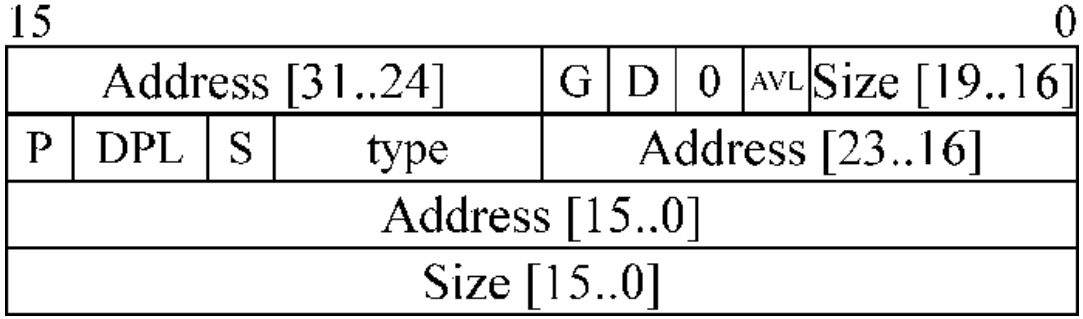
- Segment stanu zadania
  - Segment systemowy (deskryptor w GDT)
    - Selektor: TR
  - Zawiera pełny kontekst zadania:
    - Rejestry segmentowe: CS, DS, ES, FS, GS, SS
    - Rejestry adresowe: LDTR, CR3
    - Rejestry robocze: EAX, EBX, ECX, EDX, ESI, EDI, ESP, EBP, EIP, Eflags
    - Wskaźniki stosu: SS:ESP także dla „wewnętrznych” poziomów uprzywilejowania

486 – ochrona zadań

- Segment stanu zadania
  - Pozostałe pola
    - Mapa zezwoleń na dostęp do we-wy
      - *Które adresy we-wy są dostępne dla programu?*
    - Znacznik *Nested Task*, pole *Previous task*
      - Gdy zadanie jest zagnieżdżone w innym (wywołane)
        - » Np. procedura obsługo przerwania
      - Pole *Previous task* wskazuje na TSS zadania wywołującego
    - Znacznik *Debug Trap*
      - Po przełączeniu zadań wywołanie wyjątku *Trap*

486 – ochrona zadań

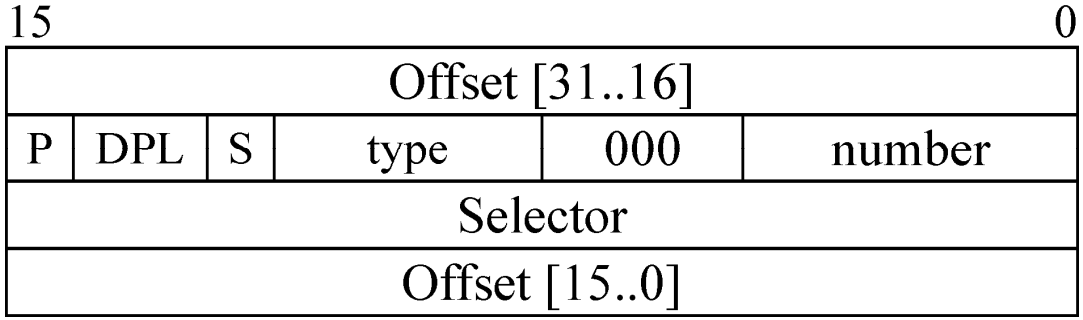
- Deskryptor segmentu systemowego
  - Np. TSS, LDT itp.



  - Typ segmentu:
    - Dostępny TSS 286
    - Zajęty TSS 286
    - Dostępny TSS 386
    - Zajęty TSS 386
    - Tablica LDT
    - Bramki wywołań

486 – ochrona zadań

- Deskryptor bramki wywołania
  - Punkt wejścia do segmentu kodu

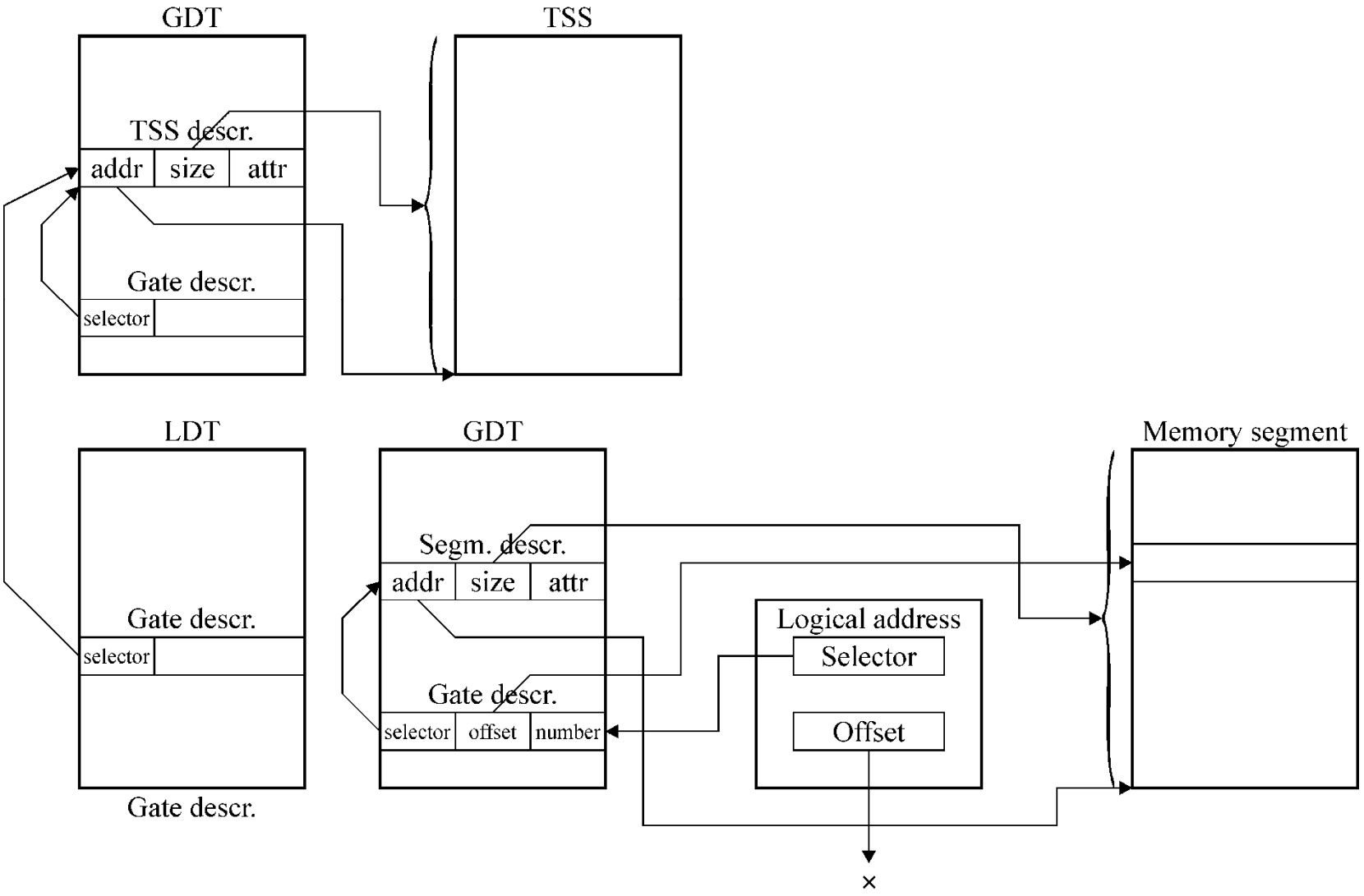


  - Typ segmentu:
    - Bramka wywołania (286/386)
    - Bramka zadania (tylko 286)
    - Bramka przerwania (286/386)
    - Bramka potrzasku (286/386)

- Pole Liczba (*Number*) tylko w bramce wywołania
- Offset nieużywany w bramce zadania
  - Selektor wskazuje na TSS
  - CS:EIP obecne w TSS

486 – ochrona zadań

- Przykłady zastosowań bramek wywołania



486 – ochrona zadań

- Przerwania, wyjątki
  - Przerwanie
    - Wytworzone poza µp
      - Asynchroniczne względem programu
      - Bez (bliskiego) związku z bieżącą instrukcją
  - Wyjątek
    - Wytworzony wewnątrz µp
      - Synchroniczny względem programu
      - (Bardzo) silny związek z bieżącą instrukcją

486 – ochrona zadań

- Przerwania, wyjątki
  - Klasy wyjątków
    - Niepowodzenie (*fault*)
      - Zgłoszony przed (pełnym) dokończeniem instrukcji
      - Instrukcję można ponowić
    - Potrzask (*Trap*)
      - Zgłoszony po zakończeniu instrukcji
      - Instrukcji nie można ponowić
    - Załamanie (*Abort*)
      - Poważny błąd
      - Instrukcja bywa nieidentyfikowalna (zatem i powtórzona)
        - » Np. niespójność struktur systemowych
        - » Sensowne wyjście → restart systemu

486 – ochrona zadań

- Przerwania, wyjątki
  - W 80486 obsługa obu typów zdarzeń jest podobna
  - Tablica wektorów przerwań
    - Tryb rzeczywisty
      - Jak w 8086, ale IDTR wskazuje na położenie i rozmiar tablicy
      - Można przesunąć w dowolne miejsce pamięci
    - Tryb wirtualny
      - IDTR określa położenie i rozmiar tablicy (max. 256 elementów)
      - Element → bramka (wywołania, przerwania, potrzasku)
  - Niektóre wyjątki zostawiają na stosie kod błędu

486 – ochrona zadań

- Przerwania, wyjątki (1)
  - 0 (fault): błąd dzielenia
  - 1 (trap): wyjątki uruchamiania
  - 2 (not an exception): NMI
  - 3 (trap): pułapka
  - 4 (trap): przepełnienie
  - 5 (fault): sprawdzenie granic
  - 6 (fault): błędny kod rozkazu
  - 7 (fault): brak koprocesora

486 – ochrona zadań

- Przerwania, wyjątki (2)
  - 8 (abort): podwójne niepowodzenie
    - Wyjątek w czasie obsługi wyjątku
  - 9 (abort): przekroczenie segmentu koprocesora
  - 10 (fault): błędny TSS
  - 11 (fault): brak segmentu
  - 12 (fault): wyjątek stosu
  - 13 (fault): naruszenie ochrony
  - 14 (fault): błąd stronicowania
  - 16 (fault): błąd koprocesora
  - 17 (fault): błąd wyrównania położenia argumentu

486 – pamięć podręczna

- Pamięć podręczna
  - Wspólna dla kodu i danych
  - Struktura
    - Logiczna
      - 128 zbiorów × 4 linie × 16 B
    - Fizyczna
      - 4 bloki × 2 KB (128 linii)
    - Znaczniki
      - 128 × 21-b znaczników dla każdego bloku 2KB

486 – pamięć podręczna

- Pamięć podręczna
  - Zapis jednoczesny (*Write Through*):
    - Trafienie: zapis do pamięci podręcznej i operacyjnej
    - Chybienie: zapis tylko do pamięci operacyjnej
  - Wyłączenie pamięci podręcznej
    - Sprzętowe: $\overline{\mathrm{KEN}}$=1 (dla wybranych adresów)
    - Programowe: PCD=1 (dla wybranych stron)
    - Całkowite:
      1. CD=NW=1
      2. flush

486 – pamięć podręczna

- Pamięć podręczna
  - Linijka całkowicie ważna lub nieważna
  - Linijkę można unieważnić (logicznie opróżnić)
  - Linijka wypełniana, gdy chybienie przy odczycie
    - Ale nie, gdy chybienie przy zapisie
  - Wypełnianie linijki
    - Gdy znaleziono linijkę nieważną
    - Gdy wszystkie ważne → Least Recently Used

486 – pamięć podręczna

- Przesyły seryjne
  - Zapis: max 32b, gdy magistrala 8/16-b
  - Odczyt: max 16B (nie więcej, niż potrzeba)
    - Adres: XXXXXXX[0-F]
  - $A_{4..31}$, M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$ stabilne podczas przesyłu
  - Start: $\overline{BRDY}$ zamiast $\overline{RDY}$
  - Pozwala na dostęp do 4 DWORD w 5 taktach
    - 2-1-1-1 zamiast 2-2-2-2

486 – pamięć podręczna

- Bufory zapisu
  - Bufor pusty, magistrala wolna → bez buforowania
  - Magistrala zajęta → buforowanie
  - Magistrala wolna → zapis z buforów
    - Kolejność taka sama, jak podczas wypełniania bufora
  - Możliwy odczyt przed zapisem
    - Gdy trafienie podczas zapisu, ale chybienie podczas odczytu