

Opracowanie pytań do obrony (EZI)

KN EZI Team

12 stycznia 2016

Spis treści

I	Zagadnienia specjalnościowe	3
1	Sterowniki mikroprocesorowe i ich zastosowania	4
1.0.1	Sterownik PLC	4
1.0.2	Regulator PID	6
1.0.3	Sterownik CNC	9
1.0.4	Kontroler PAC	9
2	Lokalne sieci komputerowe	11
3	Bazy danych i ich zastosowania	13
3.1	Bazy relacyjne	14
3.2	Z wikipedii (takie powtórzenie)	15
3.2.1	Architektura Oracle	17
4	Przetwarzanie obrazów, metody i zastosowania	19
4.1	Algorytmy przetwarzania obrazów	19
4.1.1	Operacje punktowe	20
4.1.2	Histogramy	21
4.1.3	Operacje algebraiczne	21
4.1.4	Binaryzacja	21
4.1.5	Filtracje	21
4.1.6	Decymacja i interpolacja	
	22
4.1.7	Wykrywanie krawędzi	
	22
4.2	Inne zastosowania	22
5	Miary i oceny dokładności algorytmów przybliżonych	23
6	Systemy operacyjne komputerów	26

7	Zadania optymalizacji i techniki ich rozwiązywania	31
7.0.1	Sformułowanie zadania optymalizacji	31
7.0.2	Rodzaje zadań	32
8	Systemy dynamiczne, opisy własności	36
8.1	Programowanie w systemie operacyjnym Unix	36
8.1.1	Programowanie w języku powłoki	36
8.1.2	Programowanie w C	37
8.2	Komputer, architektura i oprogramowanie	37
8.2.1	Architektury	38
8.2.2	Oprogramowanie	39
II	Zagadnienia kierunkowe	41
9	Programowanie strukturalne i obiektowe	42
9.1	Fala elektromagnetyczna: typy, parametry, właściwości	43
9.2	Tranzystory bipolarne i unipolarne: budowa, właściwości i zastosowania	45
9.2.1	Tranzystory bipolarne	45
9.2.2	Tranzystory unipolarne (polowe)	49
9.3	Systemy ciągłe i dyskretne: klasyfikacja, opis	55
9.4	Zmienna losowa: właściwości, opis	55
9.4.1	Zmienne losowe	55
9.4.2	Dystrybuanta i jej własności	56
9.4.3	Funkcja gęstości	57
9.4.4	Podstawowe parametry rozkładu zmiennej losowej	57
9.5	Ciągła, dyskretna i szybka transformata Fouriera, widmo sygnału	58
9.5.1	Ciągła transformata Fouriera	58
9.5.2	Dyskretna transformata Fouriera	58
9.5.3	Szybka transformata Fouriera	58
9.5.4	Widmo sygnału	58
9.6	Modulacje analogowe i cyfrowe	59
9.6.1	Modulacje analogowe	59
9.6.2	Modulacje cyfrowe	61
9.7	Wzmacniacze operacyjne: właściwości i zastosowania	64
9.8	Mikroprocesory: budowa, zastosowania	64
9.8.1	Architektury	64
9.8.2	Budowa	65
9.8.3	Mikroprocesor a mikrokontroler	66
9.8.4	Zastosowania	66
9.9	Sieci komputerowe: budowa, protokoły, zastosowanie	67
9.9.1	Budowa	67
9.9.2	Protokoły	70
9.9.3	Zastosowania	72

Część I

Zagadnienia specjalnościowe

Rozdział 1

Sterowniki mikroprocesorowe i ich zastosowania

Główną częścią takiego sterownika jest mikrokontroler pełniący funkcję jednostki wykonującej operacje logiczne i arytmetyczne.

1.0.1 Sterownik PLC

Programowalny sterownik logiczny PLC (ang. Programmable Logic Controller) uniwersalne urządzenie mikroprocesorowe przeznaczone do sterowania pracą maszyny lub urządzenia technologicznego. Sterownik PLC musi zostać dopasowany do określonego obiektu sterowania poprzez wprowadzenie do jego pamięci żądanego algorytmu działania obiektu - zaprogramowanie go. Cechą charakterystyczną sterowników PLC odróżniającą ten sterownik od innych sterowników komputerowych jest cykliczny obieg pamięci programu. Algorytm jest zapisywany w dedykowanym sterownikowi języku programowania. Istnieje możliwość zmiany algorytmu przez zmianę zawartości pamięci programu. Sterownik wyposaża się w odpowiednią liczbę układów wejściowych zbierających informacje o stanie obiektu i żądaniach obsługi oraz odpowiednią liczbę i rodzaj układów wyjściowych połączonych z elementami wykonawczymi, sygnalizacyjnymi lub transmisji danych. Układy I/O mogą być cyfrowe lub analogowe.

Zasada działania: przeglądanie wejść i w zależności od ich stanu oraz wprowadzonego programu użytkownika ustawianie stanu wyjść dwustanowych (zał/wył). PLC jest w swej istocie komputerem przemysłowym ze specjalnie zaprojektowaną jednostką centralną oraz rozbudowanymi układami wejść/wyjść do kontaktu z zewnętrznymi urządzeniami przemysłowymi (urządzeniami polowymi). Wcześniej działanie sterowników były realizowane przy pomocy rozbudowanej sieci elektrycznej przy zastosowaniu przekaźników.

Historia

Inżynierowie z Hydramatic Division firmy GM w roku 1968 sformułowali założenia dla sterownika programowalnego, którego główną zaletą miała być redukcja wysokich kosztów związanych z niewielką elastycznością systemów przekąźnikowych. W 1969 r. został opracowany pierwszy sterownik PLC.

Założenia sterownika

- realizacja na urządzeniach półprzewodnikowych,
- elastyczność komputera,
- praca w środowisku przemysłowym (wibracje, wysoka temperatura, zapylenie itp.),
- możliwość przeprogramowywania,
- łatwe programowanie i obsługa przez elektryków i techników.

Budowa

- jednostka centralna (CPU),
- zasilacz,
- pamięć,
- układy I/O,
- port komunikacyjny (np. PROFIBUS),
- gniazdo rozszerzenia.

Języki programowania

- **drabinkowy LD** (ang. Ladder Diagram) - logika drabinkowa, a schemat podobny do klasycznego rysunku technicznego elektrycznego,
- **blokowy FBD** (ang. Function Block Diagram) - diagram bloków funkcyjnych, sekwencji linii zawierająca te bloki,
- **strukturalny ST** (ang. Structured Text) - tekst strukturalny - język zbliżony do Pascala;
- **lista instrukcji IL** (ang. Instruction List) lista instrukcji - rodzaj assemblera,
- **sekwencyjny SFC** (ang. Sequential Function Chart) sekwencyjny ciąg bloków - sekwencja bloków programowych z warunkami przejścia.

Poziomy sygnałów cyfrowych

- 5, 12, 24, 48 VDC,
- 24, 128, 240 VAC.

Poziomy sygnałów analogowych

- 0 ... 20 mA,
- 4 ... 20 mA (najczęściej),
- 0 ... 5, 10 V,
- -10 ... 10 V.

Źródłami sygnałów analogowych są: termometry, przepływomierze, wilgotnościomierze, mierniki nacisku, potencjometry.

Analogowe sygnały wejściowe sterują: zaworami analogowymi, szybkością obrotów silników, rejestratorami.

Zadania PLC

- skanowanie wejść,
- wykonywanie instrukcji,
- ustawianie wyjść,
- samodiagnostyka.

1.0.2 Regulator PID

Regulator PID (ang. Proportional-Integral-Derivative controller) – regulator stosowany w układach regulacji składający się z trzech członów: proporcjonalnego, całkującego i różniczkującego. Najczęściej jego celem jest utrzymanie wartości wyjściowej na określonym poziomie, zwanym wartością zadaną. Jest najczęściej stosowanym regulatorem w przemyśle.

Właściwości dynamiczne idealnych regulatorów PID mogą być opisane za pomocą transmitancji operatorowej, będącej stosunkiem transformaty Laplace'a sygnału wyjściowego regulatora $U'(s)$ i transformaty Laplace'a uchybu regulacji (sygnału wejściowego) $E(s)$:

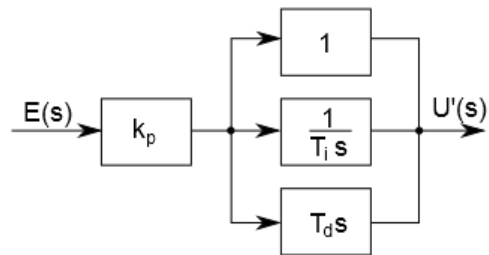
$$G_r(s) = \frac{U'(s)}{E(s)} = k_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right), \quad (1.1)$$

gdzie:

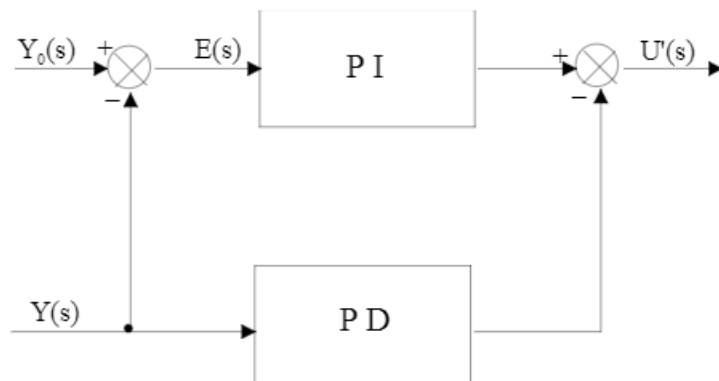
k_p - wzmocnienie członu proporcjonalnego,

T_i - czas całkowania (zdwojenia),

T_d - czas różniczkowania (wyprzedzenia).



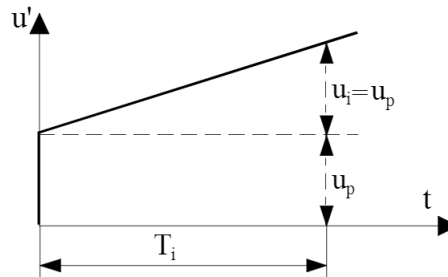
Rysunek 1.1: Schemat blokowy idealnego regulatora PID



Rysunek 1.2: Schemat blokowy mikroprocesorowego regulatora PID

Zakres proporcjonalności $X = (1/k_p) \cdot 100\%$ jest to procentowa część pełnego zakresu zmian wielkości wejściowej e , potrzebna do wywołania pełnej zmiany wielkości wyjściowej u' regulatora.

Czas zdwojenia (całkowania) T_i dotyczy regulatorów PI, których wielkość wyjściowa ma zarówno składową proporcjonalną u_p , jak i całkującą u_i . Czas zdwojenia określa się na podstawie znajomości charakterystyki czasowej skokowej regulatora. Jest to czas potrzebny na to, aby sygnał wyjściowy regulatora, będący wynikiem działania całkującego, stał się równy sygnałowi będącemu wynikiem działania proporcjonalnego (rysunek 1.3.).



Rysunek 1.3: Sposób wyznaczania czasu zdwojenia T_i

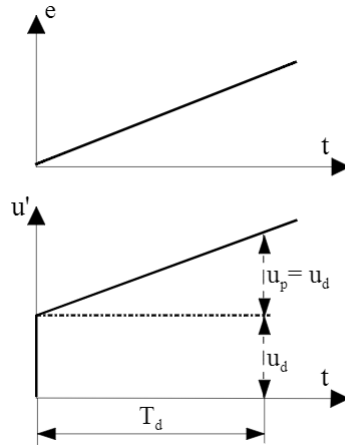
Zatem sygnał wyjściowy z regulatora PI (wypadkowy dla obu oddziaływań) po czasie T_i zwiększa dwukrotnie swoją wartość. Stąd pochodzi jego nazwa – czas zdwojenia.

Czas wyprzedzenia (różniczkowania) T_d dotyczy regulatorów PD, których wielkość wyjściowa ma zarówno składową proporcjonalną, jak i różniczkującą. Czas wyprzedzenia określa się na podstawie odpowiedzi regulatora na wymuszenie w postaci narastającego liniowo sygnału uchybu regulacji e . Jest to czas, po którym sygnał wyjściowy regulatora, związany z działaniem proporcjonalnym u_p , zrówna się z sygnałem pochodzącym od działania różniczkującego u_d (rysunek 1.4.).

Wielkości k_p (w starszych typach regulatorów X), T_i i T_d noszą nazwę nastaw regulatora. Są to parametry nastawialne, które dobieramy tak, aby:

- uzyskać odpowiednie własności dynamiczne regulatora,
- uzyskać najlepszą jakość regulacji.

Transmitancja określona wzorem (1.1) dotyczy regulatora idealnego, którego w praktyce nie daje się zrealizować. W warunkach rzeczywistych w członie różniczkującym występuje inercja, określona stałą czasową T . Stała ta nie jest nastawialna. Zatem zamiast członu $T_d \cdot s$ w transmitancji występuje człon $T_d s / Ts +$



Rysunek 1.4: Sposób wyznaczania czasu wyprzedzenia T_d

1. Transmittancja operatorowa rzeczywistego regulatora PID ma wówczas postać:

$$G_r(s) = \frac{U'(s)}{E(s)} = k_p \left(1 + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{Ts + 1} \right), \quad (1.2)$$

1.0.3 Sterownik CNC

Sterownik CNC (ang. Computer Numerical Control) to typ sterownika mikroprocesorowego, który programuje się za pomocą tzw. G code'u. Sterowniki te używane są m.in. do kontroli takich urządzeń jak: frezarki, tokarki a w szerszym zastosowaniu do sterowania robotami fabrycznymi, które pracują w tzw. trybie taśmy montażowej np. automaty składające podzespoły samochodowe.

W nowoczesnych maszynach CNC stosuje się sterowniki pracujące na bazie komputera przemysłowego IPC w technologii „PC-based Automation”. W tej technologii sterownik CNC działa programowo, a nie sprzętowo, tak jak to odbywało się w starego typu dedykowanych sterownikach. System operacyjny czasu rzeczywistego sterownika, realizuje funkcje PLC, HMI i sterowania ruchem, odpowiadając za funkcjonalność całej maszyny.

1.0.4 Kontroler PAC

Sterowniki PAC (ang. Programmable Automation Controller) zwane sterownikami automatyki są dużymi systemami sterownikowymi przeznaczonymi do obsługi złożonych procesów technologicznych. Wiele dotychczasowych sterowników PLC o największej wydajności zostało nazwanych przez producentów sterownikami PAC. Granica pomiędzy dużymi i wydajnymi sterownikami PLC a sterownikami PAC jest obecnie słabo widoczna. Współczesne procesy sterowania wykorzystują ogromne ilości sygnałów i danych, począwszy od analogowych

i cyfrowych urządzeń wejścia/wyjścia, poprzez szybkie kamery wysokiej rozdzielczości, a kończąc na wieloosiowych sterownikach ruchu. Takie zastosowania, jak: szybka produkcja, monitoring pracy maszyn w czasie rzeczywistym, sterowanie precyzyjne czy sterowanie złożonym procesem, wymagają deterministycznych systemów akwizycji, zaawansowanej analizy i algorytmów przetwarzania danych. Wyższej klasy sterowniki PLC w pewnym stopniu spełniają te wymagania. Jednakże do wydajnego przetwarzania tych danych trzeba zastosować odpowiednie środki informatyczne, m.in. procesory zmiennoprzecinkowe i duże zasoby pamięci. Sterowniki PAC integrują tego rodzaju gotowe składniki sprzętowe w ramach systemu czasu rzeczywistego, oferując w ten sposób wydajną platformę dla inżynierów automatyki.

Rozdział 2

Lokalne sieci komputerowe

Sieci lokalne LAN (ang. Local Area Network), są sieciami prywatnymi obejmującymi pojedynczy budynek lub grupę budynków w obszarze o średnicy do kilku kilometrów. Sieci te są powszechnie używane do łączenia komputerów osobistych i stacji roboczych w biurach firmy i fabrykach w celu udostępnienia zasobów (np. drukarek) i wymiany informacji.

Sieci LAN od innych typów odróżniają trzy cechy:

- rozmiary,
- technologie transmisji,
- topologia.

Sieci LAN mają ograniczone rozmiary, co oznacza, że czas przesyłu w najgorszym przypadku jest ograniczony i z góry znany. Znajomość tych granic umożliwia użycie pewnych rozwiązań, które w innych przypadkach byłyby niemożliwe do zastosowania, a jednocześnie upraszcza zarządzanie siecią.

Sieci lokalne mogą korzystać z technologii transmisji opartej na kablu, do którego podłączone są wszystkie komputery. Tradycyjne sieci LAN charakteryzują się szybkością przesyłu od 10 Mb/s do 100 Mb oraz niskim opóźnieniem. Nowsze sieci lokalne działają z szybkością do 10 Gb/s.

W rozgłoszeniowych sieciach LAN możliwe są różnorodne topologie. Przykładem jest sieć magistralowa (zbudowana z kabla liniowego). W sieci magistralowej, w każdej chwili najwyżej jedno urządzenie jest nadrzędne i ma prawo do nadawania. Potrzebny jest zatem mechanizm arbitrażu który rozstrzyga konflikt gdy dwa lub więcej urządzeń chce nadawać jednocześnie. Mechanizm arbitrażu może być scentralizowany lub rozproszony. Na przykład sieć Ethernet jest siecią rozgłoszeniową opartą na magistrali ze zdecentralizowanym sterowaniem. Komputery w tej sieci mogą nadawać w dowolnej chwili; w razie kolizji dwóch lub więcej pakietów każdy komputer oczekuje losowy odstęp czasu i ponawia próbę. Drugim typem systemu rozgłoszeniowego jest pierścień. W pierścieniu każdy bit propaguje się samodzielnie bez czekania na resztę pakietu, do którego należy.

Zazwyczaj każdy bit okrąża cały pierścień w czasie potrzebnym na transmisję kilku bitów, często zanim komplety pakiet zostanie wysłany.

Rozdział 3

Bazy danych i ich zastosowania

Baza danych – zbiór danych zapisanych zgodnie z określonymi regułami. W węższym znaczeniu jest to program (system) służący do zbierania, przetwarzania i organizowania informacji. Bazy danych pozwalają przechowywać dowolne informacje, na przykład informacje o ludziach, produktach czy zamówieniach. Dla systemów komputerowych mogą to być dane numeryczne, tekstowe, binarne, daty, lub inne dostępne typy w danym systemie. Można więc przechowywać grafikę, muzykę – np. jako pliki binarne. Kartoteka w bibliotece to również baza danych. Za bazę danych uważany może być plik tekstowy, arkusz kalkulacyjny, plik binarny. Gdy ilość danych rośnie, powstają nadmiarowe i niespójne dane. Dlatego wprowadza się różne systemy bazodanowe umożliwiające łatwiejsze zarządzanie dużą ilością danych – np. MS SQL, MySQL, Access.

Z uwagi na cenę danych współczesne systemy bazodanowe stanowią podstawę działalności wielu firm, co wymaga istnienia możliwie niezawodnych systemów baz danych.

System zarządzania bazą danych – oprogramowanie umożliwiające tworzenie oraz eksploatację bazy danych oraz jej użytkowników.

baza danych = dane + schemat bazy danych (np. relacyjny)

system bazy danych = baza danych + system zarządzania bazą

Kryteria klasyfikacji baz danych

- wielkość
- liczba odwołań
- stopień ważności informacji
- struktura informacji
- implementacja komputerowa

Bazy danych można podzielić według struktur organizacji danych, których używają:

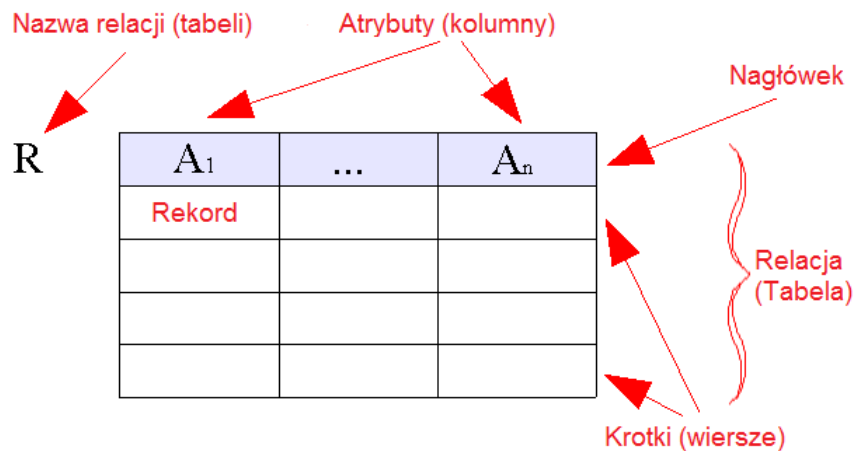
- Bazy proste: kartotekowe, hierarchiczne (struktura drzewa, np katalogi w systemie)
- Bazy złożone: relacyjne, obiektowe, relacyjno-obiektowe, strumieniowe, temporalne, nierelacyjne (NoSQL), grafowe

Z wymienionych struktur, w praktyce zdecydowanie najczęściej używane są bazy relacyjne.

3.1 Bazy relacyjne

Relacyjny model danych został zaproponowany w latach 70. Zaimplementowany w latach 80.

Opiera się on (suprise) na pojęciu relacji. Dane grupowane są w relacje reprezentowane przez tablice. Czyli tablica STUDENCI zawierająca wiersze (krotki) z danymi studentów (imię, indeks - kolumny), o odpowiednich typach - całość tworzy relację. W modelu obiektowym relacji odpowiada obiekt.



Rysunek 3.1: Relacja

Pojęcia

- klucz podstawowy (główny) - najkrótszy zbiór atrybutów (czyli kolumn) pozwalający na **jednoznaczną** identyfikację wiersza w tabeli. Max 1 na tabelę

- klucz obcy - kolumna zawierająca wartości wskazujące na *klucz główny* w innej tabeli, pozwala stworzyć związek pomiędzy tabelami. Typy związków: jeden do jednego, jeden do wielu, wiele do wielu (najczęściej realizowane za pomocą tabeli pośredniej)

Związki pomiędzy relacjami pozwalają uniknąć nadmiarowości danych i pozwalają zapewnić ich spójność. Jeżeli np. mamy tabelę studenci i samochody, relacja wiele do jednego (każdy samochód posiada klucz obcy czyli np. indeks studenta). Jeden student może posiadać więc wiele samochodów. Jeżeli chcemy sprawdzić czyj jest dany samochód to mamy klucz obcy studenta, a nie jesgo wszystkie dane - czyli wystarczy raz wpisać studenta, a zmiana jego danych następuje też w jednym miejscu. Czyli minimalna ilość danych zapewnia spójność. Zły przypadek - tabela samochody zawiera wszystkie dane studenta, on zmienia imię, trzeba aktualizować dane we wszystkich jego samochodach (i pewnie o którychś się zapomni).

Zasady projektowania schematów relacyjnych baz danych opisują **zasady normalne**, najważniejsze:

- 1NF atrybuty są atomowe (niepodzielne) - kolumna [imię,nazwisko] - należy rozdzielić na 2 osobne kolumny, jedna komórka zawiera jedną wartość, czyli nie chcemy kolumny imiona
- 2NF każda kolumna niekluczowa zależy funkcyjnie od całego klucza głównego (i tak nikt o to nie zapyta ani nikt nie wykuje, ale wstawiam)
- 3NF każda komórka w wierszu zależy jedynie od klucza głównego, [indeks], [imię]], [płeć]
Zakładając że indeks to klucz główny, a płeć jest jednak związana z imieniem to ta ka tabela nie spełni tego wymagania, należy wydzielić drugą tabelę imiona: [imię], [płeć] a studentowi zostawić indeks i klucz obcy do tabeli imiona

Do pracy z relacyjnymi bazami danych służy SQL - standardowy język zapytań, używany do tworzenia, modyfikowania baz danych oraz do umieszczania i pobierania danych z baz danych. Konkretnie: selekcje, projekcje, złączanie, usuwanie, modyfikację dodawania danych, a także operacje na schematach bazy danych - tworzenie, modyfikowanie tabel.

Zawiera on typy danych, umożliwia tworzenie funkcji, procedur składowanych.

3.2 Z wikipedii (takie powtórzenie)

Relacyjne bazy danych (jak również przeznaczony dla nich standard SQL) oparte są na kilku prostych zasadach:

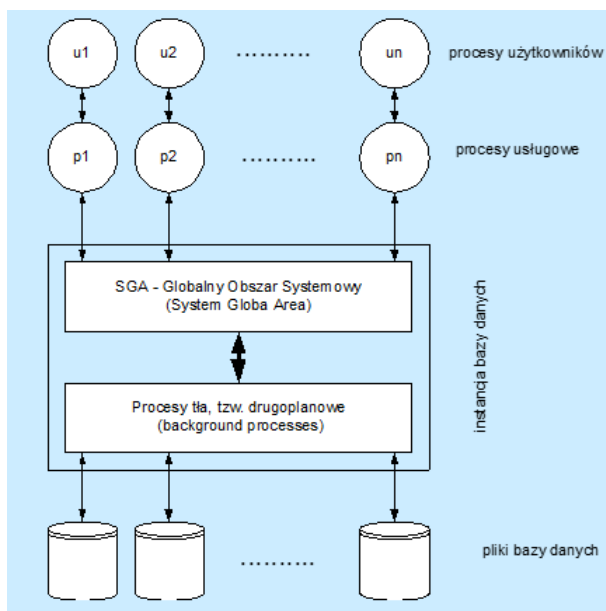
1. Wszystkie wartości danych oparte są na prostych typach danych.

2. Wszystkie dane w bazie relacyjnej przedstawiane są w formie dwuwymiarowych tabel (w matematycznym żargonie noszących nazwę „relacji”). Każda tabela zawiera zero lub więcej wierszy (w tymże żargonie – „krotki”) i jedną lub więcej kolumn („atrybutów”). Na każdy wiersz składają się jednakowo ułożone kolumny wypełnione wartościami, które z kolei w każdym wierszu mogą być inne.
3. Po wprowadzeniu danych do bazy, możliwe jest porównywanie wartości z różnych kolumn, zazwyczaj również z różnych tabel, i scalanie wierszy, gdy pochodzące z nich wartości są zgodne. Umożliwia to wiązanie danych i wykonywanie stosunkowo złożonych operacji w granicach całej bazy danych.
4. Wszystkie operacje wykonywane są w oparciu o algebrę relacji, bez względu na położenie wiersza tabeli. Nie można więc zapytać o wiersze, gdzie ($x=3$) bez wiersza pierwszego, trzeciego i piątego. Wiersze w relacyjnej bazie danych przechowywane są w porządku zupełnie dowolnym – nie musi on odzwierciedlać ani kolejności ich wprowadzania, ani kolejności ich przechowywania.
5. Z braku możliwości identyfikacji wiersza przez jego pozycję pojawia się potrzeba obecności jednej lub więcej kolumn niepowtarzalnych w granicach całej tabeli, pozwalających odnaleźć konkretny wiersz. Kolumny te określa się jako „klucz podstawowy” (ang. primary key) tabeli.

Zastosowania

- systemy bankowe (bankomat)
- systemy masowej obsługi (hipermarket)
- rezerwacja biletów lotniczych
- telefonia komórkowa (sms)
- Dziekanat Wydziału Elektroniki
- toto-lotek
- policja (ewidencja przestępców, rejestr samochodów)
- rejestry sądowe, księgi wieczyste
- ankiety internetowe
- sklepy internetowe
- gry internetowe
- system audiotele
- biblioteka PWr.

3.2.1 Architektura Oracle



Rysunek 3.2: Architektura systemu

Rodzaje plików

- pliki konfiguracyjne (init files) — najważniejszy z nich to plik parametrów (tzw. parameter file, w skrócie pfile) o nazwie init<SID>.ora, gdzie <SID> to unikalny (najczęściej 4-literowy) identyfikator instancji na danej maszynie, plik jest tekstowy i zawiera informacje o ustawieniach przy starcie bazy, a także o nazwie i położeniu plików kontrolnych
- pliki kontrolne (control files) — binarne, bardzo ważne, zawierają informacje o położeniu WSZYSTKICH plików bazy danych wraz z datą i godziną ich zamknięcia
- pliki danych (data files) — przechowują dane w postaci binarnej, z danych w nich zawartych korzysta się za pośrednictwem procesów serwera (tła)
- pliki dziennika powtórzeń (redo-log files) - rejestrują wszystkie operacje wykonane na bazie (na wypadek awarii), mogą być aktywne lub zarchiwizowane
- pliki śladu (trace files) — zawierają informacje o uszkodzeniach i o błędach procesów tła oraz procesów użytkowników

- pliki kodu (code files) — wszelkie kody źródłowe, skrypty SQL itp

SGA (ang. System Global Area). Składa się z:

- bufora danych, który: Zawiera dane załadowane z dysku, okresowo zapisane (Dirty blocks), przechowuje informacje o zmianach, ma strukturę cykliczną.
- bufora dziennika powtórzeń, który przechowuje informację o zmianach, ma strukturę cykliczną.
- obszaru współdzielonego

Procesy tła

- DBWR — database writer — zapisuje DIRTY BLOCKS na dysk
- LGWR — log writer — zapisuje zmiany do loga
- CKPT — checkpoint — sygnalizuje tzw. punkt kontrolny + aktualizuje nagłówki plików kontrolnych
- SMON — system monitor — odtwarza instancję, scala wolne obszary na dysku
- PMON — process monitor — zwalnia zasoby procesu użytkownika, który się „zawiesił”
- ARCH — archiwizator — opcjonalny, spowalnia pracę, umożliwia skonfigurowanie serwera lustrzanego

Stany bazy danych

- IDLE - nieczynna, pliki zamknięte, procesy tła nie działają
- NOMOUNT - stan po odczytaniu pfile-a, zainicjowaniu SGA i uruchomieniu procesów tła, stan służący do tworzenia nowej bazy danych
- MOUNT - stan po odczytaniu plików kontrolnych i otwarciu połączenia z plikami danych i log-ami
- OPEN - stan otwarcia bazy, dane są dostępne dla użytkowników (w wyjątkowych sytuacjach stan można uruchomić w tzw. trybie RESTRICTED, wtedy dostęp do bazy mają tylko administratorzy)

Rozdział 4

Przetwarzanie obrazów, metody i zastosowania

Reprezentacja obrazu Obraz cyfrowy powstaje poprzez kwantyzację i próbkowanie obrazu analogowego. Wynikiem jest uporządkowana siatka kwadratów, z których każdy zawiera informację o obrazie wejściowym jak np. średnia jasność w obszarze. Każdy z kwadratów siatki zwany jest pikselem, a ich liczba i wielkość są podstawowymi wartościami opisującymi matryce służące do akwizycji obrazu jak CCD lub CMOS.

Rozróżniamy obrazy jednobarwne - *monochromatyczne*, binarne, barwne.

Cyfrowy zapis obrazu polega na zdefiniowaniu przestrzeni barw w jakiej zapisujemy obraz (RGB, CMYK, inne), głębi obrazu - czyli bitów na kanał co określa ilość możliwych do przedstawienia barw. Przykładowo:

typowa kamera internetowa - ramka obrazu składa się z trzech kanałów - obrazów monochromatycznych, dla których zakres wartości przypisywanych pikselom należy do przedziału 0-255. Kanały reprezentują kolory z palety barw RGB, obraz końcowy powstaje poprzez złączenie wszystkich kanałów. Zapewnia to 24-bitową głębię kolorów (3 kanały po 8 bitów), tzn. pozwala reprezentować 2^{24} kolorów. Tryb ten nazywany jest *true color*. 4 kanałem często używanym np. w formacie png może być kanał A *Alpha* odpowiadający za (nie)-przezroczystość.

4.1 Algorytmy przetwarzania obrazów

Pojęcie **przetwarzania obrazów** można rozumieć rozmaicie. W wąskim sensie oznacza operacje, które przetwarzają jedne obrazy w inne - zarówno dane wejściowe jak i wyjściowe mają formę obrazu. W tym sensie zawierają się:

- akwizycja obrazów - wprowadzenie obrazów, zwłaszcza cyfrowych do systemów technicznych (sorki, brzydka definicja),
- reprezentacja obrazu i modelowanie - dobór reprezentacji obrazu w systemie tech., by odpowiadała ona charakterowi obrazu i przeznaczeniu re-

prezentacji, również modele matematyczne obrazów. Głębia, przestrzeń barw

- polepszanie obrazów - wytwarzanie obrazów subiektywnie ocenianych przez człowieka jako lepsze
- odzyskiwanie, restauracja obrazów - usuwanie zniekształceń (geometrycznych), zakłóceń (szumy, rozmycia)
- kompresja obrazów - oczywiste
- znakowanie wodne - umieszczanie i odczytywanie dodatkowych, często niewidocznych informacji
- prezentacja, wyświetlanie obrazów

Z tego punktu widzenia grafika komputerowa, animacje komputerowe stanowią oddzielną grupę metod.

W szerszym sensie do przetwarzania obrazów zalicza się **analizę obrazów**, w której dane wejściowe odpowiadają obrazom, natomiast dane wyjściowe mają inny charakter. A analizą obrazów związane jest **rozpoznawanie wzorców** którego celem jest identyfikacja pewnych struktur w obrazach.

3 kategorii metod przetwarzania stanowi **analiza obrazów**:

- analiza ogólnych cech obrazu lub jego fragmentu - np. analiza cech ilościowych wartości pikseli, barwy punktów
- ekstrakcja cech - np. krawędzi, struktur geometrycznych
- segmentacja - podział obrazu na spójne obszary o podobnych cechach (ruch, kolor)
- analiza tekstur - wydzielenie np. drewna, tkaniny i analiza cech
- analiza sceny i zdarzeń - określanie zależności przestrzennych i czasowych pomiędzy obiektami, analiza ruchu obiektów, zdarzeń itp.

4.1.1 Operacje punktowe

Należą do I grupy metod (obraz w obraz). Przekształcają każdy z pikseli wejściowych $u(n_1, n_2)$ w piksel $y(n_1, n_2)$ według:

$$y(n_1, n_2) = F[u(n_1, n_2)]$$

Gdzie n_1, n_2 - indeks poszczególnego piksela, np. współrzędne x,y, trzeci parametr (n_3) może być kanałem. Przykład - dodanie +70 do wartości każdego piksela przyciemni obraz kolorowy jako całość, przy obrazie binarnym zamiana 0 na 1 i odwrotnie - negatyw obrazu. Wszystkie piksele obliczane są niezależnie, co pozwala zrównoleglać te algorytmy. Przykłady:

- dodanie stałej C $y(n_1, n_2) = u(n_1, n_2) + C$

- inwersja obrazu $y(n_1, n_2) = 255 - u(n_1, n_2)$ (dla 8-bitowego obrazu)

Operacje punktowe pozwalają wykorzystywać tablicowanie, (*look-up-table* - LUT), zamiast wyliczać każdy piksel przechowujemy w pamięci zbiór wszystkich wartości, np. 40 zamieniamy na 70, 41 na 71... dzięki czemu wystarczy sprawdzić i wstawić wartość zamiast liczyć.

Wszystkie operacje punktowe można wykonywać z nasyceniem lub bez, dodanie 100 do 200 w 8 bitowej głębi (max 255) może dać czarny (255) przy nasyceniu (operacja nieliniowa!), lub 45 (czyli jasny) przy „zawinięciu” się wartości.

4.1.2 Histogramy

Wyliczenie histogramu pozwala „zobaczyć” ile pikseli jakiego koloru występuje w obrazie, np. dominują piksele czerwone nad zielonymi, albo jasne nad ciemnymi. Pozwala ocenić kontrast - równomierny histogram oznacza wiele detali w obrazie (dużo różnych pikseli), w innym przypadku obraz jest np. niedoświetlony. Histogram pozwala również opisać operacje punktowe. Dodanie stałej to nic innego jak przesunięcie histogramu.

4.1.3 Operacje algebraiczne

Operacje punktowe działające na wielu obrazach, przykładowo mając kilka identycznych obrazów, z różnym szumem można je dodać i uśrednić eliminując szum.

4.1.4 Binarizacja

- progowanie, zmiana obrazu na binarnym, według jakiegoś progu, np. $u < 100 \geq 0$, $u \geq 100 \geq 1$.

4.1.5 Filtracje

Algorytmy usuwające zakłócenia, szumy. Wartość piksela obliczana jest na podstawie jego i otoczenia - o jego wielkości i kształcie decyduje maska, np. prostokątna o wymiarach 5x5. Przykładowe filtry:

- **uśredniający** - najprostszy, liczy średnią wartość pikseli pod maską, rozmywa obraz (ale usuwa szumy)
- **Gaussa** - rozmywający, ale według rozkładu normalnego. Czyli średnia ważona
- **bilateralny** - Gaussa z uwzględnieniem różnicy kolorów (jak zupełnie inne piksele to nie uśrednia), usuwa szum, zachowuje krawędzie więc bardzo przydatne przed np. wykrywaniem krawędzi
- **medianowy** - filtr nieliniowy, wartość piksela równa jest medianie pikseli pod maską, suwa zakłócenia impulsowe z obrazu (*pieprz i sól*), losowe bardzo jasne/ciemne piksele na obrazie

4.1.6 Decymacja i interpolacja

Zmniejszenie/zwiększenie rozdzielczości obrazu. Interpolację można opisać jako wstawienie zer w miejsca nowych próbek, a następnie wygładzenie obrazu filtrem dolnoprzepustowym.

4.1.7 Wykrywanie krawędzi

Rodzina algorytmów krawędziujących, np. na podstawie gradientu czyli skokowych zmian wartości pikseli. Przydatne przy algorytmach identyfikujących, pozwalają łatwiej ocenić kształt.

4.2 Inne zastosowania

Wszelkie algorytmy identyfikujące, twarz, rękę, samochód. Działają na podstawie wykrycia kształtu i/lub koloru. Okrągłe i cieliste - raczej głowa, kolorowe i kwadratowe - pewnie samochód. (sorki, chyba za późno to piszę).

Rozdział 5

Miary i oceny dokładności algorytmów przybliżonych

Algorytmy aproksymacyjne – algorytmy służące do znajdowania przybliżonych rozwiązań problemów optymalizacyjnych. Stosuje się je zwykle do rozwiązywania problemów, dla których nie są znane szybkie algorytmy znajdujące rozwiązanie dokładne, na przykład dla problemów NP-zupełnych.

Istotą algorytmu aproksymacyjnego, tym co odróżnia go od algorytmu heurystycznego, jest związana z każdym takim algorytmem informacja o koszcie zwracanego rozwiązania w stosunku do rozwiązania optymalnego. Mianowicie koszt rozwiązania zwróconego przez algorytm aproksymacyjny jest nie większy (w przypadku problemu minimalizacyjnego) albo nie mniejszy (w przypadku problemu maksymalizacyjnego) od rozwiązania optymalnego pomnożonego przez pewną stałą.

Oszacowania jakości algorytmów aproksymacyjnych

Załóżmy, że mamy do czynienia z problemem optymalizacyjnym, w którym każde potencjalne rozwiązanie ma dodatni koszt i, że chcemy znaleźć rozwiązanie prawie optymalne. Zależnie od problemu rozwiązanie optymalne może być zdefiniowane jako to o maksymalnym możliwym koszcie lub o minimalnym; zadanie może polegać na maksymalizacji albo na minimalizacji.

Mówimy, że algorytm aproksymacyjny dla danego problemu ma ograniczenie względne $p(n)$, jeśli dla dowolnych danych wejściowych rozmiaru n koszt C rozwiązania konstruowanego przez ów algorytm szacuję się z dokładnością do czynnika $p(n)$ przez koszt C' rozwiązania optymalnego:

$$\max\left(\frac{C}{C'}, \frac{C'}{C}\right) \leq p(n)$$

Definicja ta stosuję się zarówno do problemów minimalizacji, jak i maksymalizacji. Dla problemu maksymalizacji $0 < C \leq C'$, a współczynnik C'/C

określa ile razy koszt rozwiązania optymalnego jest większy od kosztu rozwiązania przybliżonego. Podobnie dla problemu minimalizacji $0 < C' \leq C$, a współczynnik C/C' określa ile razy koszt rozwiązania przybliżonego jest większy od kosztu rozwiązania optymalnego. Ponieważ zakładamy, że wszystkie rozwiązania mają dodatni koszt, współczynniki te są zawsze dobrze określone. Ograniczenie względne algorytmu aproksymacyjnego nigdy nie jest mniejsze niż 1, ponieważ nierówność $C/C' < 1$ implikuje $C'/C > 1$. Ograniczeniem względnym algorytmu optymalnego jest 1, a algorytm aproksymacyjny o dużym ograniczeniu względnym może dać rozwiązanie znacznie gorsze niż optymalne.

Czasami wygodniej operować pojęciem błędu względnego. Dla dowolnych danych wejściowych błąd względny definiuję się jako:

$$\frac{|C - C'|}{C'}$$

gdzie, jak poprzednio C' jest kosztem rozwiązania optymalnego, a C to koszt rozwiązania uzyskanego za pomocą algorytmu aproksymacyjnego. Błąd względny jest zawsze nieujemny. Dla algorytmu aproksymacyjnego $\epsilon(n)$ jest ograniczeniem błędu względnego, jeśli:

$$\frac{|C - C'|}{C'} \leq \epsilon(n)$$

Z powyższych definicji wynika, że ograniczenie błędu względnego można oszacować przez funkcję ograniczenia względnego

$$\epsilon(n) \leq p(n) - 1$$

(Dla problemu minimalizacyjnego jest to równość, podczas gdy dla problemu maksymalizacyjnego mamy $\epsilon(n) = (p(n) - 1)/p(n)$; spełniona jest więc nierówność, ponieważ $p(n) \geq 1$).

Dla wielu problemów skonstruowano algorytmy aproksymacyjne o stałym, niezależnym od n ograniczeniu względnym. W takich przypadkach będziemy pisać p lub ϵ wskazując w ten sposób na niezależność od n .

Dla innych problemów informatykom nie udało się zaprojektować żadnego wielomianowego algorytmu aproksymacyjnego o stałym ograniczeniu względnym. Wszystko co można wówczas robić to potraktować ograniczenie względne jako funkcję rosnącą wraz z rozmiarem danych wejściowych. Przykładem takiego problem jest problem pokrycia zbioru.

Dla niektórych problemów NP- zupełnych istnieją algorytmy aproksymacyjne za pomocą których można uzyskiwać coraz mniejsze ograniczenie względne (lub, równoważnie, malejący błąd względny) kosztem dłuższego czasu obliczeń. Inaczej mówiąc, zachodzi odwrotnie proporcjonalna współzależność między czasem obliczeń a jakością aproksymacji. Przykładem może być problem sumy podzbioru. Sytuacja jest na tyle ważna, że zasługuje na własną nazwę.

Schemat aproksymacyjny dla problemu optymalizacyjnego to algorytm aproksymacyjny, otrzymujący na wejściu nie tylko komplet danych opisujących problem, ale także wartość $\epsilon > 0$ taką, że dla każdego ustalonego ϵ schemat ten jest algorytmem aproksymacyjnym o ograniczeniu błędu względnego ϵ . Mówimy, że schemat aproksymacyjny jest wielomianowym schematem aproksymacji, jeśli dla dowolnego ustalonego $\epsilon > 0$ działa on w czasie wielomianowym ze względu na rozmiar n jego danych wejściowych.

Czas działania wielomianowego schematu aproksymacji nie powinien też rosnąć zbyt gwałtownie, kiedy zmniejsza się ϵ . Najlepiej, gdyby zmniejszenie ϵ o stały czynnik nie powodowało wzrostu czasu obliczeń potrzebnych do osiągnięcia dostatecznego przybliżenia o więcej niż stały czynnik. Innymi słowy, chcielibyśmy aby czas obliczeń było wielomianem ze względu na n , ale również, ze względu na $1/\epsilon$.

Mówimy, że schemat aproksymacji jest w pełni wielomianowym schematem aproksymacji, jeśli czas jego działania jest wielomianem zarówno ze względu na $1/\epsilon$ jak i rozmiar n danych wejściowych, gdzie ϵ jest ograniczeniem błędu względnego schematu. Czas działania schematu mógłby na przykład wynosić $(1/\epsilon)^2 \cdot n^3$. Dla takiego schematu dowolne zmniejszenie o stały czynnik ograniczenia błędu względnego dają się osiągnąć za pomocą odpowiedniego zwiększenia o stały czynnik czasu obliczeń.

Rozdział 6

Systemy operacyjne komputerów

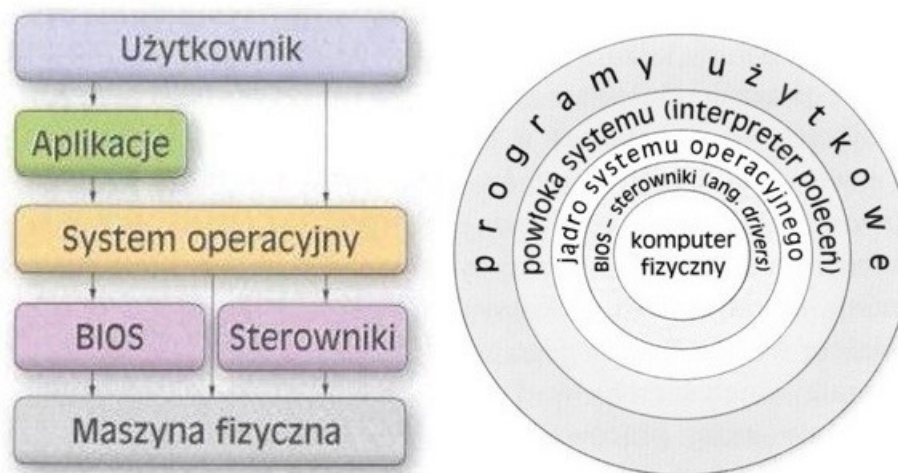
System operacyjny (ang. Operating System) jest środowiskiem programów tworzących główną platformę programową umożliwiającą działanie zainstalowanego w systemie oprogramowania. OS nadzoruje pracę wszystkich uruchomionych procesów oraz urządzeń komputera. Pomimo iż swą pracę wykonuje przeważnie w tle i sam w sobie nie tworzy z komputera w pełni funkcjonalnego narzędzia jednak bez niego komputer jest kompletnie bezużyteczny. System operacyjny zainstalowany na dysku twardym komputera decyduje jakie oprogramowanie może zostać uruchomione pod jego kontrolą, wpływa na bezpieczeństwo danych, realizuje połączenia do sieci nadzorując podrzędne systemy pracujące na innych komputerach, określa kompatybilność wobec innych systemów, decydując o funkcjonalności stabilności pracy. Niestety nie istnieje system idealny, każdy ma swoje zalety i wady.

Budowa i zadania poszczególnych warstw OS

Podczas pracy OS komunikuje się zazwyczaj z elektroniką komputera poprzez BIOS i sterowniki choć jest możliwa również jego bezpośrednia komunikacja ze sprzętem co zostało pokazane na rysunku 6.1. System operacyjny można przedstawić w postaci modelu warstwowego.

Główne zadania stawiane przed systemami operacyjnymi to:

- zarządzanie zasobami komputera - systemy operacyjne starają się oraz optymalizują wykorzystanie określonych urządzeń, które wchodzi w skład zestawu komputerowego; specjalne moduły zwane sterownikami, które składają się na system operacyjny udostępniają aplikacjom spójne metody programowania urządzeń (interfejsy), co gwarantuje współdziałanie każdego nowego urządzenia z oprogramowaniem (jeżeli producent dostarczy właściwy sterownik)



Rysunek 6.1: Warstwowy model systemu

- gromadzenie oraz zarządzanie danymi - systemy operacyjne wyposażone są w moduły obsługujące system plików, czyli struktury umieszczone na dyskach, pomagające w logiczny sposób uporządkować dane, grupując je w katalogi i pliki
- maszyny wirtualne - systemy operacyjne udostępniają aplikacjom tzw. maszyny wirtualne, czyli uproszczone obrazy maszyn, na których pracują aplikacje; system udostępnia aplikacjom szczegóły dna temat komputera oraz rozszerzenia ułatwiające pracę (np. zasoby udostępniane poprzez sieć aplikacje widzą tak, jakby znajdowały się one na dysku lokalnym; aplikacja, która korzysta z takiego zasobu nie obsługuje pracy sieciowej, więc aby umożliwić jej dostęp do zasobu, system operacyjny symuluje, jest zasób ten jest lokalny udostępniając go dla aplikacji)
- wielozadaniowość - na pojedynczym komputerze funkcjonować może wiele aplikacji w jednym czasie; każda aplikacja otrzymuje swoją maszynę wirtualną i może pracować jakby była pojedynczym programem działającym na maszynie; dzięki takiej właściwości systemów operacyjnych nie ma potrzeby przystosowywania aplikacji, by mogła dzielić się daną maszyną z innymi aplikacjami
- interakcja z użytkownikiem - rolę tę spełnia najbardziej zewnętrzna warstwa systemu operacyjnego zwana powłoką (ang. shell), która pozwala użytkownikowi uruchamiać aplikacje; w systemach graficznych do powłoki zaliczają się także typowe elementy interfejsu, z których korzysta aplikacja, jak kontrolki czy okna dialogowe

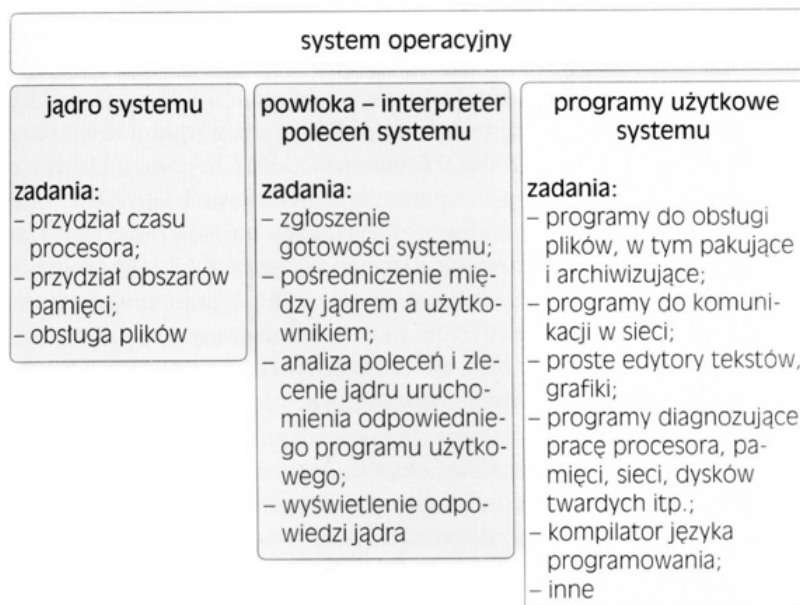
- komunikowanie się z innymi komputerami - jest to jeden z najistotniejszych elementów systemów operacyjnych; dzięki modułom obsługi sieci można uzyskać dostęp do Internetu, do dysków komputerów stojących w sąsiednim pomieszczeniu czy sieciowych urządzeń peryferyjnych jak drukarki czy skanery

Najważniejszymi cechami, które decydują o użyteczności danego systemu operacyjnego są:

- prostota instalacji oraz użytkowania systemu
- współpraca z innymi systemami, czyli możliwość odczytu i zapisu danych na partycjach z innych systemów a także współpraca oraz wymiana danych między komputerami w sieciach lokalnych i Internecie
- zgodność sprzętowa (instalację na konkretnej maszynie utrudnia czasami brak właściwych sterowników dla określonych urządzeń)
- wymiana danych, czyli możliwość przeglądania oraz wymiany dokumentów pomiędzy różnymi aplikacjami pracującymi pod kontrolą różnych systemów
- możliwość pracy sieciowej (wygoda podczas przeglądania zasobów sieciowych, wymiany protokołów, itp.)
- cena
- liczba aplikacji, które działają w danym systemie (nawet najlepiej pracujący system będzie niemal bezużyteczny, jeżeli oferta oprogramowania, które współpracuje z tym systemem będzie niewielka)
- lokalizacja (możliwość komunikacji użytkownika z systemem w ojczystym języku)

Schemat z rysunku 6.1. przedstawia ogólną budowę systemu operacyjnego. Podstawową częścią OS jest jądro systemu (kernel). Jądro odpowiedzialne jest za pracę systemu i wykonywanie wszystkich jego zadań. Jądro poprzez sterowniki i BIOS komunikuje się z elektroniką komputera. Aby użytkownik mógł komunikować się z jądrem system operacyjny posiada powłokę (Shell lub też interpreter). Powłoka systemowa jest to program pełniący rolę pośrednika pomiędzy całym systemem operacyjnym a użytkownikiem. Powłoki mogą być tekstowe lub graficzne.

Systemy operacyjne podzielić można ze względu na:



Rysunek 6.2: Składniki systemu operacyjnego i ich zadania

Sposób komunikacji systemu z użytkownikiem:

- systemy tekstowe - komunikacja przebiega przy pomocy komend wprowadzanych z linii poleceń (DOS, CP/M)
- systemy graficzne - komunikacja odbywa się przy pomocy graficznych symboli (okienek oraz ikon); obsługa systemu polega na manipulacji przy pomocy myszy bądź klawiatury symbolami odpowiadającym określonym zadaniom (MC Windows, MacOS)

Architekturę systemu:

- monolityczne - jednozadaniowe systemy posiadające najprostszą strukturę, gdzie w danym czasie może być realizowane tylko jedno zadanie
- warstwowe - posiadające hierarchiczną strukturę poleceń systemowych; możliwa jest realizacja wielu zadań jednocześnie (np. nadzorowanie procesu drukowania podczas edycji tekstu)
- klient/serwer - systemy posiadające bardzo rozbudowaną strukturę, nadzorujące podrzędne systemy zainstalowane na komputerach w sieci; systemy te postrzegają aplikacje jako klientów, korzystających z usług serwerów; aplikacja „klient” komunikuje się z serwerem przez jądro systemowe, natomiast każdy serwer działa we własnej, chronionej i wydzielonej przestrzeni

adresowej w pamięci operacyjnej, gdzie jest odizolowany od innych zadań; systemy klient/serwer realizują swe zadania na trzy sposoby:

- wszelkie aplikacje uruchamiane są na serwerze, a wyniki prezentowane u "klienta"
- serwer dostarcza zasobów dla aplikacji, które uruchamiane są po stronie klienta
- wszelkie komputery współdzielą ze sobą na zasadzie równy z równym (ang. peer to peer), wykorzystując wzajemnie swoje zasoby

Przykłady systemów:

- Windows
- Linux
- Unix
- OS X/ Mac OS X
- Android

Rozdział 7

Zadania optymalizacji i techniki ich rozwiązywania

7.0.1 Sformułowanie zadania optymalizacji

Wektor zmiennych decyzyjnych x :

$$x = [x_1, x_2, \dots, x_n]^T,$$

gdzie:

n - ilość zmiennych decyzyjnych.

Funkcja celu (funkcja kryterialna) $f(x)$:

$$f(x) : R^n \longrightarrow R^1$$

oraz m funkcji ograniczeń $g_i(x)$:

$$g_i(x) : R^n \longrightarrow R^1 \quad \text{dla } i = 1, \dots, m$$

Zadanie optymalizacji polega na znalezieniu wektora zmiennych decyzyjnych x , należącego do zbioru rozwiązań dopuszczalnych \mathbf{X} w postaci:

$$X = \{x | g_i(x) \leq 0, \quad i = 1, \dots, m\}$$

takiego, że dla wszystkich $x \in X$

$$f(\hat{x}) \leq f(x).$$

Co jest równoznaczne zapisowi:

$$\min_{x \in X} f(x) = f(\hat{x}).$$

7.0.2 Rodzaje zadań

Zadanie programowania liniowego PL

$$\max f(x) = c^T x$$

przy ograniczeniach:

$$A_1 x \leq b_1$$

$$A_2 x \geq b_2$$

$$x \geq 0$$

$$\dim x = n, \dim c = n$$

Macierze A_1, A_2 odpowiadają za współczynniki w m_1 i m_2 ograniczeniach
Wektory b_1, b_2 odpowiadają za prawe strony ograniczeń

Przypadki szczególne rozwiązania zadania programowania liniowego (PL)

- Istnieje jedno rozwiązanie optymalne zadani PL.
- Zadanie PL jest zadaniem nieograniczonym – brak rozwiązania.
- W zadaniu PL zbiór rozwiązań dopuszczalnych jest pusty – brak rozwiązania.
- W zadaniu PL wszystkie zmienne lub ich część przyjmuje
- wartości rzeczywiste.
- Zadanie PL posiada nieskończoną liczbę rozwiązań.

Metody rozwiązywania:

- simpleks
- dwufazowy simpleks
- dualny simpleks

Zadanie liniowego programowania całkowitoliczbowego PCL

Zagadnieniem liniowym całkowitoliczbowym nazywamy zadanie optymalizacji liniowej, w którym wszystkie zmienne lub ich część są liczbami całkowitymi.

$$\max_{x \in X} c^T x \quad \text{dla } x \in X \subset R^n$$

dla każdego

$$1 \leq i \leq t \quad x_i \in \mathbb{Z}$$

Metody uproszczone:

- Przegląd zupełny zbioru rozwiązań dopuszczalnych X
- Zaokrąglenie rozwiązania optymalnego zadania PL do zmiennych całkowitych

Metody złożone:

- Metoda odcięć Gomory'ego
- Metoda podziału i ograniczeń
- Metody przybliżone = metody heurystyczne

Zadanie programowania nieliniowego PN

$$\min_{x \in X} f(x) = f(\hat{x})$$

przy ograniczeniach:

$$X = \{x | g_i(x) \leq 0, i = 1, \dots, m\}$$

Zadanie programowania nieliniowego polega na znalezieniu wektora zmiennych decyzyjnych \hat{x} , należącego do zbioru rozwiązań dopuszczalnych X w postaci: takiego, że dla

$$x \in X$$

$$f(\hat{x}) \leq f(x)$$

Nieliniowe zadanie optymalizacji statycznej bez ograniczeń Funkcja celu $f(x)$:

$$f(x) : R^n \longrightarrow R^1$$

Zadanie optymalizacji polega na znalezieniu wektora zmiennych decyzyjnych x , należącego do zbioru rozwiązań dopuszczalnych R^n takiego, że dla każdego $x \in R^n$

$$f(\hat{x}) \leq f(x)$$

Co jest równoznaczne zapisowi:

$$\min_{x \in R^n} f(x) = f(\hat{x})$$

Iteracyjne algorytmy optymalizacji:

- Algorytmy optymalizacji w kierunku
- Algorytmy optymalizacji bez ograniczeń
- Algorytmy optymalizacji z ograniczeniami

Algorytm optymalizacji lokalnej - przemierzanie obszaru rozwiązań dopuszczalnych w poszukiwaniu ekstremum funkcji celu według iteracyjnego schematu.

Algorytmy bezgradientowe:

- Algorytm Nelder'a-Meade'a (Matlab - funkcja fminsearch)
- Algorytm Gauss'a-Seidla
- Algorytm Powella

Algorytmy gradientowe:

- Algorytm największego spadku
- Zmodyfikowany algorytm Newtona (Matlab – wersja metody QuasiNewton - funkcja fminunc)
- Algorytm Zangwilla
- Algorytm Fletchera-Reeves'a
- Algorytm Polaka-Ribiera
- Algorytm Fletchera-Powella-Davidona

Nieliniowe zadanie optymalizacji statycznej z ograniczeniami

Znaleźć wektor rozwiązań optymalnych \hat{x} , *taki*, że :

$$f(\hat{x}) = \min_{x \in X} f(x)$$

gdzie:

$$X = \{x : g_i(x) \leq 0, \quad i = 1, \dots, m\}$$

$$f : X \subset \mathbb{R}^n \longrightarrow \mathbb{R}^1$$

oraz

$$g_i : X \subset \mathbb{R}^n \longrightarrow \mathbb{R}^1 \text{ dla każdego } i = 1, \dots, m$$

Ograniczenie aktywne

Dla \hat{x} ograniczenie przyjmuje postać:

$$g_i(\hat{x}) = 0$$

$$g_i(\hat{x} + \tau d) \leq 0, \text{ dla każdego } i \in A(\hat{x})$$

przy czym $\tau \in [0, \sigma]$

Przypadki:

1. Jedno ograniczenie aktywne – rozwiązanie optymalne na brzegu ograniczenia aktywnego.
2. Wszystkie ograniczenia aktywne - rozwiązanie optymalne na przecięciu ograniczeń aktywnych.
3. Brak ograniczeń aktywnych – rozwiązanie optymalne wewnątrz obszaru rozwiązań dopuszczalnych (rozwiązanie optymalne zadania bez ograniczeń spełnia ograniczenia problemu.

Rozdział 8

Systemy dynamiczne, opisy własności

8.1 Programowanie w systemie operacyjnym Unix

8.1.1 Programowanie w języku powłoki

Powłoka (ang. shell) jest programem umożliwiającym pracę z systemem UNIX. Jej nazwa wywodzi się z graficznej prezentacji systemu, w której jądro i procesy obsługi wejścia i wyjścia są otoczone właśnie przez powłokę. Jej funkcją jest odseparowanie użytkownika od bezpośredniego dostępu do jądra systemu i umożliwienie łatwej i wszechstronnej pracy. Podstawowym zadaniem powłoki jest przyjmowanie poleceń użytkownika i uruchamianie stosownych programów. Do dalszych funkcji powłoki należy konstruowanie parametrów dla poleceń oraz zarządzanie sposobem wykonywania programów procesów. Następnym, bardziej skomplikowanym krokiem jest tworzenie funkcji i procedur powłoki (skryptów) pozwalających na łączenie wielu poleceń w jedno, umożliwiając pisanie całych programów dla powłoki. Powłoka jest programem, z którym użytkownik systemu UNIX spotyka się na samym początku. Jej uruchomienie następuje automatycznie, bezpośrednio po zalogowaniu (zgłoszeniu) użytkownika w systemie. Gdy widzimy na ekranie znak „\$”, oznacza to, że powłoka jest już uruchomiona i oczekuje na podawanie poleceń.

Powłoki UNIX

Bourne shell (sh)

- nie umożliwia ona operowania na liczbach całkowitych bez tworzenia nowego procesu,
- plikiem wykonywalnym powłoki na większości systemów Unix jest `/bin/sh`,
- podstawowa powłoka w każdym systemie typu Unix.

Bash

W rozwinięciu Bourne-Again Shell (born again shell) - wywodzi się od powłoki Bourne'a sh i był pisany w 1987 r. Jest nadal rozwijany.

- umożliwia wykonanie obliczeń za pomocą wyrażeń w podwójnych nawiasach ((...)) oraz składni \$[...].

Korn shell (ksh)

- kompatybilna wstecz z sh,
- zawiera wiele elementów z csh np. historię komend.

C shell (csh)

- o składni języka C,
- pochodzi od sh,
- ulepszenia: aliasy i historia komend,
- jej następcą jest tcsh,
- rzadko wykorzystywana.

Polecenia powłoki

Wydanie polecenia w powłoce wiąże się z podaniem ciągu znaków. Składa się z następujących elementów:

<nazwa polecenia> -<opcje> <parametry>

Nazwą polecenia jest nazwa jakiegokolwiek programu, funkcji powłoki lub funkcji wewnętrznej powłoki. Opcje to najczęściej zestaw pojedynczych liter określających sposób wykonania programu. Parametry to informacje dla programu (funkcji) co właściwie ma wykonać.

8.1.2 Programowanie w C

8.2 Komputer, architektura i oprogramowanie

Komputer [ang. < łac. computare 'rozważać', 'obliczyć'], elektroniczna maszyna cyfrowa, urządzenie elektroniczne przeznaczone do przetwarzania informacji (danych) przedstawionych w postaci cyfrowej, sterowane programem zapisanym w pamięci. Pojęcie komputera obejmuje obecnie zarówno komputery zaprogramowane na stałe, używane jako automaty sterujące, np. w urządzeniach gospodarstwa domowego, jak i komputery uniwersalne, dające się dowolnie zaprogramować.

8.2.1 Architektury

Model von Neumanna a model współczesnego komputera

John von Neumann opublikował w 1945 propozycję opracowania nowego komputera. Zasadniczą nowością była koncepcja przechowywania programu w pamięci. Komputer miał składać się z czterech bloków funkcjonalnych:

- pamięć przechowująca program do wykonania i dane dla niego;
- jednostka arytmetyczno-logiczna zawierająca rejestry AC (accumulator), MQ (multiplier-quotier), MBR (memory buffer register);
- jednostka sterująca zawierająca licznik programu PC (program counter), rejestr adresowy MAR (memory address register) i pomocnicze rejestry IBR (instruction buffer register), IR (instruction register);
- urządzenia wejścia-wyjścia

Składniki współczesnego komputera:

- (mikro)procesor, zawierający co najmniej jedną jednostkę arytmetyczno-logiczną, jednostkę sterującą i rejestry;
- pamięć operacyjna;
- urządzenia wejścia-wyjścia (klawiatura, mysz, karta graficzna, pamięci dyskowe itp.);
- układ bezpośredniego dostępu do pamięci (ang. DMA – direct memory access);
- układ przerwań.

Urządzenia wejścia-wyjścia same też mogą być skomplikowanymi układami mikroprocesorowymi i zawierać w swoim wnętrzu mikroprocesor, pamięć operacyjną, układy wejścia-wyjścia itd. Układ DMA jest specjalizowanym procesorem odciążającym procesor główny przy operacjach przesyłania dużych bloków danych między pamięcią operacyjną a układami wejścia-wyjścia. Komunikacja pomiędzy poszczególnymi składnikami odbywa się za pomocą szyn (zwanych też magistralami): danych, adresowych, sygnałów sterujących.

Architektury typu Princeton i Harvard

Architektura typu Princeton posiada wspólną hierarchię pamięci programu i danych, jak opisano to w modelu von Neumanna. Architektura typu Harvard polega na rozdzieleniu pamięci programu od pamięci danych. Stosowana jest dla zwiększenia wydajności w pamięciach podręcznych oraz w systemach wbudowanych (np. sterownikach urządzeń AGD, gdzie kod programu nie zmienia się przez całe życie urządzenia lub zmienia się rzadko)

Pamięć operacyjna Pamięć operacyjna (ang. internal memory, primary storage) – pamięć komputerowa adresowana i dostępna bezpośrednio przez procesor, a nie za pośrednictwem urządzeń wejścia-wyjścia. W pamięci tej mogą być umieszczane rozkazy procesora (program) dostępne bezpośrednio dla jego jednostek wykonawczych.

Urządzenia wejścia-wyjścia Urządzenie wejścia-wyjścia, urządzenie we/wy, urządzenie I/O służy do komunikacji systemu komputerowego z jego użytkownikiem lub innym systemem przetwarzania danych. Urządzenie wejścia-wyjścia służy często do zamiany wielkości fizycznych na dane przetwarzane przez system lub odwrotnie. Np. mysz komputerowa przetwarza ruch ręki, odbiornik GPS aktualne położenie geograficzne, a monitor komputera przetwarza dane komputerowe na obraz.

DMA - Direct Memory Access DMA jest to metoda, która umożliwia urządzeniom I/O wysłanie oraz odbieranie danych bezpośrednio do lub z pamięci operacyjnej, pomijając CPU w celu przyspieszenia tych operacji oraz odciążenia procesora który uczestniczy w procesie w niewielkim stopniu. Ma on za zadanie odpowiednio programować kontroler DMA który zajmuje się transferem danych.

System przerwań

Przerwanie sygnał powodujący zmianę przepływu sterowania, niezależnie od aktualnie wykonywanego programu. Pojawienie się przerwania powoduje wstrzymanie aktualnie wykonywanego programu i wykonanie przez procesor kodu procedury obsługi przerwania (ang. interrupt handler). Procedura ta wykonuje czynności związane z obsługą przerwania i na końcu wydaje instrukcję powrotu z przerwania, która powoduje powrót do programu realizowanego przed przerwaniem.

- Przerwania sprzętowe (maskowalne, niemaskowalne)
- Przerwania programowe
- Praca krokowa
- Wyjątki
- Tablica przerwań

8.2.2 Oprogramowanie

Oprogramowanie (ang. software) – całość informacji w postaci zestawu instrukcji, zaimplementowanych interfejsów i zintegrowanych danych przeznaczonych dla komputera do realizacji wyznaczonych celów. Celem oprogramowania jest przetwarzanie danych w określonym przez twórcę zakresie.

Rodzaje oprogramowania

- oprogramowanie systemowe - realizujące funkcje konieczne dla działania systemu
- oprogramowanie do tworzenia oprogramowania
- biblioteki programistyczne
- oprogramowanie użytkowe

Część II

Zagadnienia kierunkowe

Rozdział 9

Programowanie strukturalne i obiektowe

Programowanie strukturalne – jest to „podparadygmat” programowania proceduralnego. Opiera się na tworzeniu programów z kilku dobrze zdefiniowanych funkcji takich jak instrukcja warunkowa if-then-else i pętle, za to bez skoków (go to). Proponowane jest używanie tylko trzech struktur sterujących:

- **Sekwencja lub konkatencja** – wykonywanie instrukcji w określonej kolejności.
- **Wybór** – wykonywanie jednej z kilku instrukcji zależnie od stanu programu. Przykładem jest if-then-else i switch/case.
- **Iteracja** - przetwarzanie instrukcji tak długo, jak długo spełniony (lub niespełniony) jest dany warunek. Np. while, for. Te struktury stosuje się do „małych” bloków programu złożonych z elementarnych instrukcji tj. podstawień, wywołań procedur/funkcji, instrukcji IO itd... Duże bloki powinny być rozbite na mniejsze (funkcje, procedury) tak aby rozumieć poszczególne fragmenty bez rozumienia całości. Podział ten ma również wpływ na jakość kodu, ponieważ procedura/funkcja może być używana wielokrotnie bez niepotrzebnego powielania kodu.

Programowanie obiektowe – W programowaniu obiektowym program to zbiór porozumiewających się ze sobą obiektów, czyli jednostek zawierających określone dane i umiejących wykonywać na nich określone operacje. Najważniejsze są tu dwie cechy: po pierwsze, powiązanie danych (czyli stanu) z operacjami na nich (czyli poleceniami) w całość, stanowiącą odrębną jednostkę — obiekt; Programowanie obiektowe ułatwia, pisanie, konserwację, testowanie i wielokrotne użycie programów lub ich fragmentów.

Podstawowe założenia paradygmatu obiektowego:

- **Abstrakcja** – każdy obiekt w systemie służy jako model abstrakcyjnego „wykonawcy” który może wykonywać pracę (metody), opisywać swój stan, zmieniać swój stan oraz komunikować się z innymi obiektami bez ujawniania jego implementacji.
- **Hermetyzacja** – ukrywanie implementacji, enkapsulacja. Zapewnia, że obiekt nie może zmienić stanu wewnętrznego innych obiektów w nieoczekiwany sposób. Każdy obiekt prezentuje innym obiektom swój interfejs który określa dopuszczalne metody współpracy.
- **Polimorfizm** – wielopostaciowość, referencje mogą dotyczyć obiektów różnego typu. Wywołanie metody dla referencji spowoduje zachowanie odpowiednie dla typu obiektu danej referencji. Jeśli dzieje się to w trakcie wykonywania programu to nazywa się to wiązaniem dynamicznym.
- **Dziedziczenie** – porządkuje i wspomaga polimorfizm. Umożliwia definiowanie i tworzenia specjalizowanych obiektów na podstawie bardziej ogólnych (od ogółu do szczegółu). Dzięki temu nie powiela się kodu oraz można redefiniować szczególne parametry bez redefiniowania parametrów ogólnych.

Języki wspierające programowanie obiektowe m.in.: C#, Java, C++, JavaScript, Objective-C

Źródło : materiały PW

9.1 Fala elektromagnetyczna: typy, parametry, właściwości

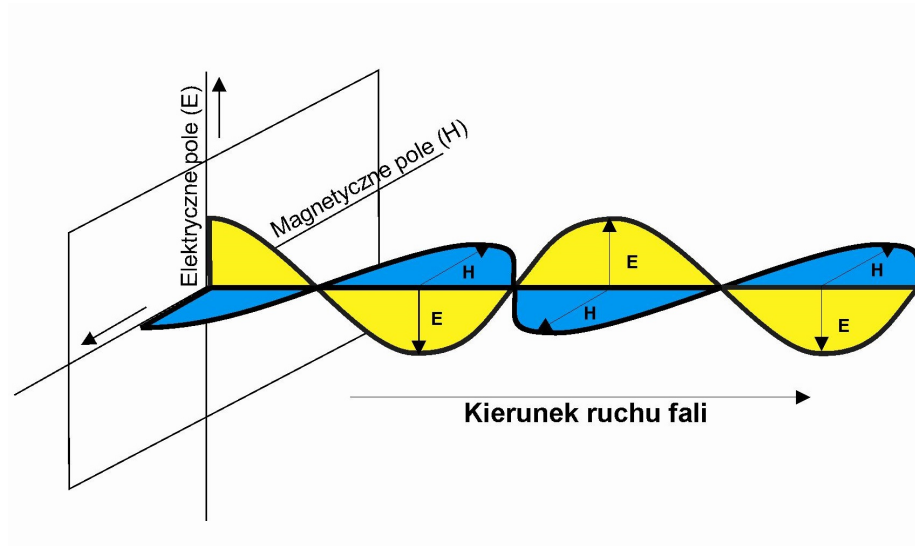
Pole elektryczne – każdy ładunek wytwarza w przestrzeni wokół siebie pole elektryczne, siła elektrostatyczna działająca na dowolny ładunek jest wywołana polem elektrycznym, wytworzonym przez inne ładunki w miejscu w którym znajduje się rozważany ładunek.

Pole magnetyczne – wytwarzany przez magnes, jest to pole wektorowe tak jak w przypadku pola elektrycznego. W elektromagnesie pole magnetyczne wytwarzane jest za pomocą przepływu prądu przez cewkę nawiniętą na metalowy rdzeń z żelaza. O sile pola decyduje wartość natężenia prądu. Magnesy trwałe zawierają cząstki elementarne (elektrony) które wytwarzają własne pole magnetyczne. W niektórych materiałach pola magnetyczne elektronów sumują się wytwarzając wokół nich wypadkowe pole magnetyczne.

Fala elektromagnetyczna – jest to rozchodząca się z prędkością c fala pola elektrycznego i magnetycznego. Fale elektromagnetyczne można podzielić na:

- Fale stojące (punkty o jednakowej fazie nie przemieszczają się) np. wnęka rezonansowa

- Bieżące (punkty o jednakowej fazie poruszają się) – rozchodzące się wzdłuż linii przesyłowej lub wolnej przestrzeni



Rysunek 9.1: Fala elektromagnetyczna

Cechy

- Wektory \vec{E} i \vec{B} są zawsze prostopadłe do kierunku rozchodzenia się fali. Zatem fale elektromagnetyczna jest falą poprzeczną.
- Iloczyn wektorowy $\vec{E} \times \vec{B}$ zawsze wyznacza kierunek rozchodzenia się fali.
- Natężenie pola elektrycznego i indukcja pola magnetycznego zmieniają się zawsze sinusoidalnie. Wektory pól zmieniają się z taką samą częstością a ich oscylacje są zgodne w fazie.

Rozważając powyższe cechy, przyjmując, że fala rozchodzi się w dodatnim kierunku osi OX. Wektor natężenia pola elektrycznego wykonuje oscylacje równoległe do osi OY a wektor indukcji równoległe do osi OZ (w prawoskrętnym układzie współrzędnych). Można zapisać następujące równania:

$$E(x, t) = E_m \cdot \sin(kx - \omega t) - \text{składowa elektryczna}, \quad (9.1)$$

$$B(x, t) = B_m \cdot \sin(kx - \omega t) - \text{składowa magnetyczna}, \quad (9.2)$$

gdzie:

E_m i B_m - amplitudy fali pola elektrycznego i magnetycznego,

ω - częstotliwość kątowna wyrażona przez wektor falowy ($\omega = ck$),
 k - stała propagacji ($k = 2\pi/\lambda$),
 λ - długość fali ($\lambda = cT$),
 T - okres drgań.

Prędkość fali elektromagnetycznej:

$$c = \frac{\omega}{k} = \frac{1}{\sqrt{\mu_0 \varepsilon_0}} = 3.0 \cdot 10^8 \frac{m}{s}, \quad (9.3)$$

gdzie:

μ_0 - przenikalność magnetyczna próżni ($4\pi \cdot 10^{-7} \text{ H/m}$),
 ε_0 - przenikalność elektryczna próżni ($8.854 \cdot 10^{-12} \text{ F/m}$).

Energia fali elektromagnetycznej – energia pojedynczego kwantu jest zależna tylko od częstotliwości fali f i wynosi:

$$E = hf, \quad (9.4)$$

gdzie:

h - stała Plancka ($6.626 \cdot 10^{-34} \text{ J} \cdot \text{s}$).

Wektor Poyntinga – strumień energii przenoszonej przez falę elektromagnetyczną w każdym punkcie przestrzeni określa wektor Poyntinga zdefiniowany, jako:

$$\vec{S} = \frac{1}{\mu_0} \vec{E} \times \vec{B}. \quad (9.5)$$

Pęd i ciśnienie fali elektromagnetycznej – biegnąca fala elektromagnetyczna niesie ze sobą pęd równy:

$$\vec{p} = \frac{W}{c} \hat{k}, \quad (9.6)$$

gdzie:

W - energia niesiona przez falę,

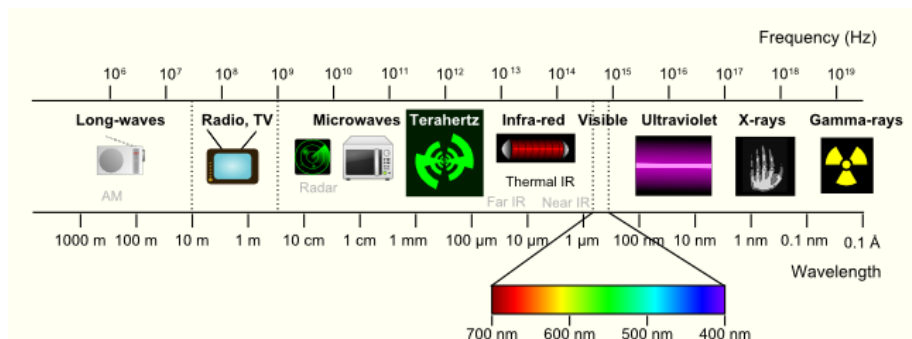
c - prędkość światła,

\hat{k} - wektor jednostkowy w kierunku rozchodzenia się fali.

9.2 Tranzystory bipolarne i unipolarne: budowa, właściwości i zastosowania

9.2.1 Tranzystory bipolarne

Tranzystor bipolarny – półprzewodnikowy element elektroniczny o trzech elektrodach. Jest to najważniejszy przykład elementu aktywnego, czyli urządze-



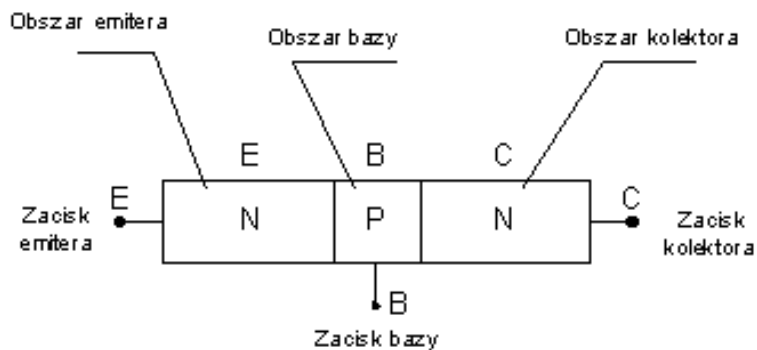
Rysunek 9.2: Widmo promieniowania elektromagnetycznego

nia, które może wytwarzać na wyjściu sygnał o mocy większej niż moc sygnału wejściowego. Służy również do przełączania sygnału.

Budowa

Tranzystor bipolarny składa się z trzech obszarów półprzewodnika o przeciwnym typie przewodnictwa, co powoduje powstanie dwóch złączy: p-n i n-p.

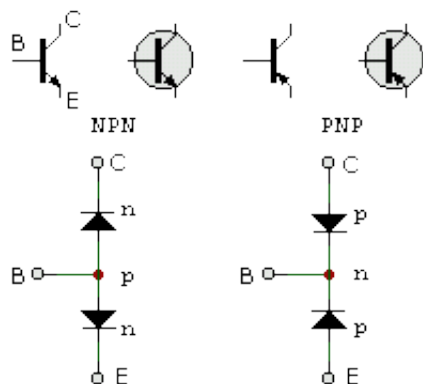
p – półprzewodnik niedomiarowy gdyż przeważają nośniki typu dziuowego
n – półprzewodnik nadmiarowy gdyż przeważają nośniki typu elektronowego



Rysunek 9.3: Budowa tranzystora bipolarnego (NPN)

Diodowy schemat zastępczy nie odzwierciedla w pełni jego działania, daje jednak pewien pogląd na napięcia występujące między elektrodami.

Główną cechą charakterystyczną tranzystora jest to, że prąd kolektora I_C



Rysunek 9.4: Diodowe schematy zastępcze tranzystorów bipolarnych

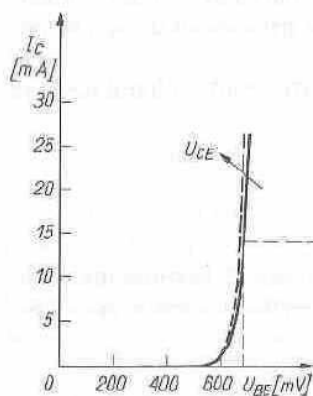
jest proporcjonalny do prądu bazy I_B . Stosunek:

$$\beta = \frac{I_C}{I_B}$$

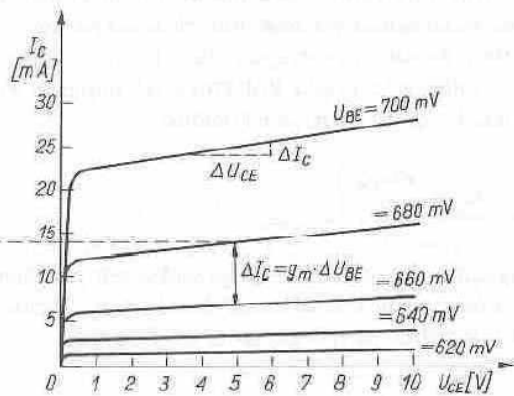
wielkosygnalowym współczynnikiem wzmocnienia prądowego.

Dalsze rozważania dotyczą tranzystorów typu pnp, w przypadku npn należy zmienić znak wszystkich prądów i napięć na przeciwny.

Po doprowadzeniu U_{BE} i dokonaniu pomiaru I_C w funkcji napięcia wyjściowego U_{CE} . Stopniowe zwiększanie napięcia wejściowego daje charakterystykę wyjściową.



Rys. 4.5. Charakterystyka przejściowa

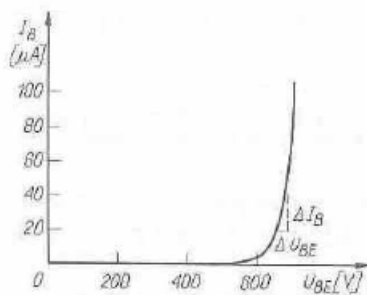


Rys. 4.6. Charakterystyka wyjściowa

Rysunek 9.5: Charakterystyki: przejściowa i wyjściowa tranzystora bipolarnego

Na podstawie charakterystyki wyjściowej można zauważyć, że powyżej pew-

nego napięcia prąd kolektora prawie nie zależy od U_{CE} . Napięcie, przy którym następuje zagięcie charakterystyk nosi nazwę napięcia nasycenia kolektor-emiter U_{CEsat} . Drugą ważną właściwością jest fakt, że do wywołania względnie dużej zmiany prądu kolektora wystarczy niewielka zmiana napięcia wejściowego. Ta zmiana, czyli odstęp między charakterystykami, silnie rośnie przy zwiększaniu prądu kolektora. Zmianę tę można dobrze zauważyć na charakterystyce przejściowej z rysunku 9.5.

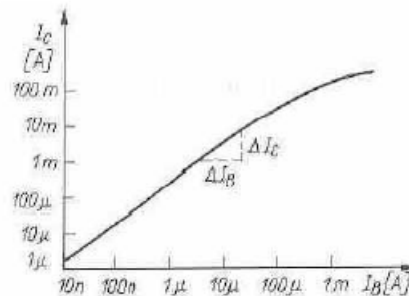


Rys. 4.7. Charakterystyka wejściowa

Rysunek 9.6: Charakterystyka wejściowa tranzystora bipolarnego

Tranzystor w przeciwieństwie do lampy elektronowej nie da się sterować bezprądowo, widać to na powyższej charakterystyce. Wzmocnienie prądowe nie jest stałe, lecz zależy od prądu kolektora.

Prąd kolektora w pierwszym przybliżeniu jest proporcjonalny do prądu bazy. Widać to na rysunku 4.8. Stosunek I_C do I_B nazywa się statycznym współczynnikiem wzmocnienia prądowego.



Rys. 4.8. Wykres typowej zależności prądu kolektora od prądu bazy dla tranzystora małej mocy

Podsumowując, właściwości tranzystorów bipolarnych:

- prąd kolektora jest proporcjonalny do prądu bazy

- powyżej pewnego napięcia prąd kolektora prawie nie zależy od U_{CE}
- do wywołania względnie dużej zmiany prądu kolektora wystarczy niewielka zmiana napięcia wejściowego
- tranzystorów bipolarnych nie da się sterować bezprądowo
- wzmacnienie prądowe nie jest stałe, lecz zależy od prądu kolektora.

Zastosowania:

Wzmacniacz – tranzystor pracujący w stanie aktywnym może być wykorzystany do budowy układu będącego wzmacniaczem natężenia prądu elektrycznego. Małe zmiany prądu elektrycznego płynącego w obwodzie bazy powodują duże zmiany prądu płynącego w obwodzie kolektora. W zależności od konstrukcji układu można uzyskać wzmacnienie prądu, napięcia lub obu tych wielkości.

Przełącznik – przy pracy tranzystora jako przełącznik wykorzystuje się przejście między stanem nasyconym (tranzystor włączony) a zatkanym (tranzystor wyłączony). Taki tryb pracy tranzystora jest stosowany w niektórych układach impulsowych oraz cyfrowych.

9.2.2 Tranzystory unipolarne (polowe)

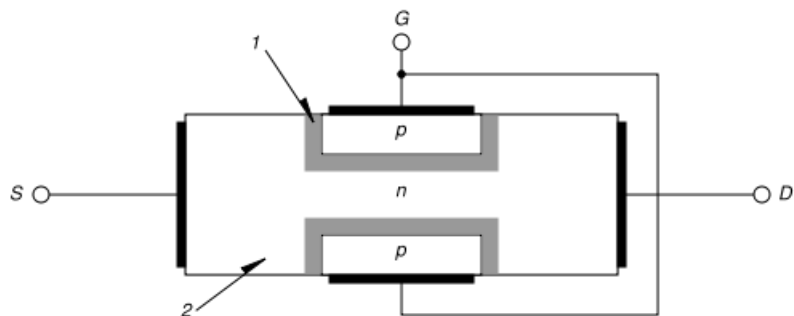
Tranzystory unipolarne (polowe) – są elementami półprzewodnikowymi, trójelektrodowymi, które w przeciwieństwie do normalnych tranzystorów bipolarnych są sterowane polem elektrycznym tzn. bez poboru mocy. Działanie tranzystorów polowych opiera się na sterowaniu przepływem prądu przez kanał za pomocą pola elektrycznego wytwarzanego przez napięcie doprowadzane do elektrody nazywanej bramką. Nie ma tu żadnych przewodzących złącz, więc do bramki nie wpływa ani z niej nie wypływa prąd. Podobnie jak w przypadku tranzystorów bipolarnych, istnieją tranzystory polowe o dwóch różnych rodzajach przewodnictwa: z kanałem typu n (przewodnictwo elektronowe) oraz z kanałem typu p (przewodnictwo dziurowe). Tranzystory polowe mogą być wykonane z dwoma różnymi rodzajami bramek (mamy więc tranzystory złączowe i z izolowaną bramką) oraz mogą różnić się sposobem domieszkowania materiału półprzewodnikowego tworzącego kanał (tranzystory ze zubożaniem i ze wzbogacaniem kanału).

Budowa JFET (tranzystora polowego złączowego)

Budowa tranzystora JFET została przedstawiona na rysunku 9.7.

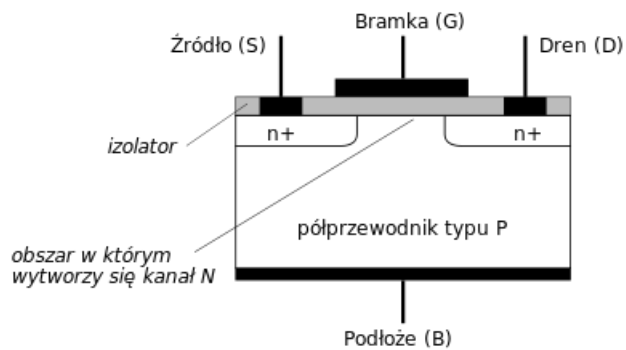
Budowa MOSFET (tranzystora polowego z izolowaną bramką)

Najważniejszą cechą tranzystora polowego jest brak prądu bramki. Wynikająca z tego duża wartość rezystancji wejściowej, przekraczająca $10^{14} \Omega$ jest



Rysunek 9.7: Budowa JFET

podstawą wielu zastosowań. Budowa MOSFET'a została ukazana na rysunku 9.8.



Rysunek 9.8: Budowa MOSFET

Rodzaje tranzystorów polowych


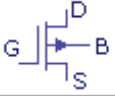
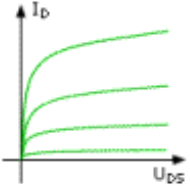
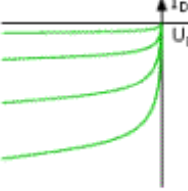
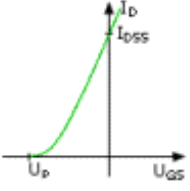
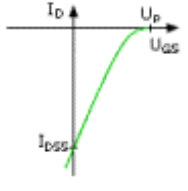
Klasyfikacja pod względem polaryzacji napięć wejściowych i wyjściowych (ze źródłem dołączonym do masy).

Właściwości tranzystorów polowych

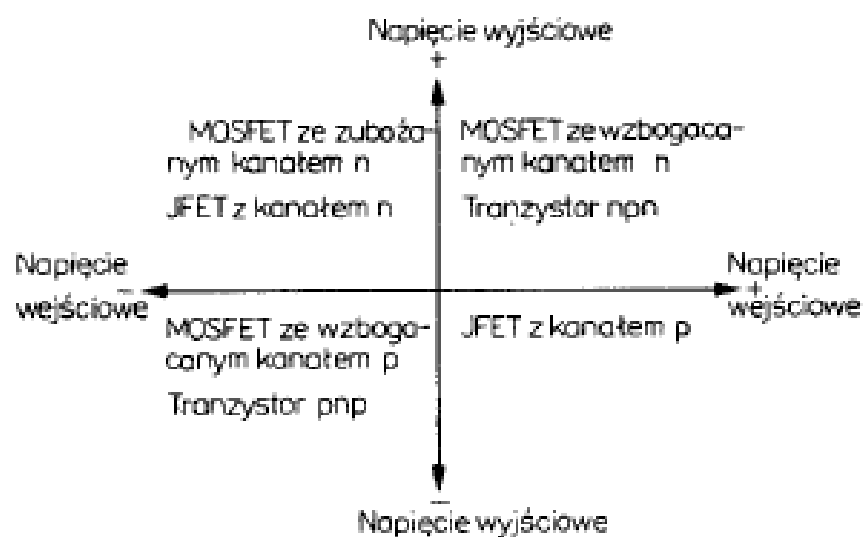
Jak wynika z powyższych rysunków w normalnym użyciu znajdują się pięć typów tranzystorów polowych. Jednakże, niekoniecznie trzeba pamiętać właściwości każdego z nich gdyż wszystkie zachowują się podobnie.

Po pierwsze, dla źródła dołączonego do masy, każdy tranzystor polowy jest wprowadzany w stan przewodzenia („włączany”) przez taką zmianę napięcia

złączowe	
kanał typu n	kanał typu p
Wzmacniacze zbudowane z elementów dyskretnych. Analogowe układy scalone.	Wzmacniacze zbudowane z elementów dyskretnych. Analogowe układy scalone.

z izolowaną bramką	
z kanałem zubożanym	
kanał typu n	kanał typu p
	
	
	
Wzmacniacze w.cz. zbudowane z elementów dyskretnych. Cyfrowe układy scalone.	Wzmacniacze w.cz. zbudowane z elementów dyskretnych. Cyfrowe układy scalone.

z kanałem wzbogacającym	
kanał typu n	kanał typu p
Wzmacniacze mocy zbudowane z elementów dyskretnych. Cyfrowe układy scalone.	Wzmacniacze mocy zbudowane z elementów dyskretnych. Cyfrowe układy scalone.



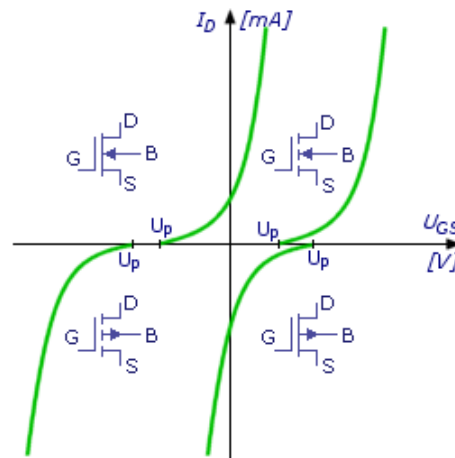
bramki, aby wartość tego napięcia dążyła do wartości napięcia zasilającego dren w warunkach aktywnej pracy tranzystora.

Po drugie, ze względu na prawie symetryczną konstrukcję tranzystora poleowego, zarówno dren jak i źródło mogą pracować, jako rzeczywiste źródło (wyjątek stanowią tranzystory MOS mocy, w których podłoże jest połączone ze źródłem wewnątrz obudowy).

Podobnie jak tranzystor n-p-n, tranzystor polowy charakteryzuje się dużą przyrostową wartością impedancji obwodu drenu. Przejawia się to stałą wartością prądu drenu, niezależną od wartości napięcia U_{DS} jeśli tylko wartość tego napięcia jest większa od powiedzmy, 1 V. Im większa jest wartość napięcia między bramką a źródłem tym większa jest wartość prądu drenu. W obszarze nasycenia, czyli w normalnym obszarze pracy FET-a prąd drenu nie jest szybkozmienna funkcją napięcia U_{GS} . Prąd drenu jest proporcjonalny do $(U_{GS} - U_T)^2$, gdzie U_T jest napięciem progowym czyli napięciem U_{GS} dla którego zaczyna płynąć prąd drenu.

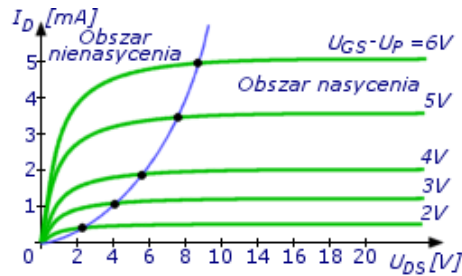
Podstawowe charakterystyki:

Przejęciowa – zależność prądu drenu (I_D) od napięcia bramka-źródło (U_{GS}) przy stałym napięciu dren-źródło (U_{DS}).



Rysunek 9.9: Charakterystyka przejściowa tranzystorów unipolarnych

Wyjściowa – zależność prądu drenu (I_D) od napięcia dren-źródło (U_{DS}), przy stałym napięciu bramka-źródło (U_{GS}). Cały obszar charakterystyki wyjściowej można podzielić na dwie części: obszar nasycenia i obszar nienasycenia (liniowy). Na rysunku 9.10 obszary te są rozdzielone niebieską linią, której kształt przypomina parabolę.



Rysunek 9.10: Charakterystyka wyjściowa tranzystora unipolarnego

W zakresie liniowym (nienasylenia) tranzystor unipolarny zachowuje się jak rezystor półprzewodnikowy. Prąd I_D ze wzrostem napięcia U_{DS} wzrasta w przybliżeniu liniowo. W zakresie nasycenia napięcie U_{DS} bardzo nieznacznie wpływa na prąd drenu, natomiast bramka zachowuje właściwości sterujące.

9.3 Systemy ciągłe i dyskretne: klasyfikacja, opis

9.4 Zmienna losowa: właściwości, opis

9.4.1 Zmienne losowe

Zmienna losowa X to taka funkcja $X : \Omega \rightarrow \mathbb{R}$, dla którego dla dowolnego borelowskiego zbioru $B \subset \mathbb{R}$ zbiór:

$$\{\omega : X(\omega) \in B\} = \{X \in B\} \in F, \quad (9.7)$$

tzn. zbiór $X \in B$ jest zdarzeniem losowym.

Innymi słowy, jest to taka funkcja X na zbiorze zdarzeń elementarnych o wartościach liczbowych, dla której określone są (teoretycznie) prawdopodobieństwa przyjmowania przez X wartości z każdego dowolnego zakresu.

Zmienna losowa to funkcja przekształcająca wynik eksperymentu losowego na liczbę rzeczywistą.

Pojęcie zmiennej losowej jest bardzo użyteczne, pozwala na abstrahowanie od postaci przestrzeni zdarzeń a operowanie wyłącznie na liczbach.

Wyobraźmy sobie rzut kostką sześciocinną. Jest to przykład eksperymentu losowego, czyli eksperymentu, dla którego wiemy jakie sytuacje mogą się wydarzyć, ale nie wiemy która się wydarzy. Możliwe wyniki eksperymentu są najróżniejsze, np. kostka będzie turlała się przez 2 minuty i wypadnie sześć oczek, kostka spadnie ze stołu i wypadnie jedno oczko itp.

Zmienna losowa to funkcja przedstawiająca wynik eksperymentu w postaci liczby rzeczywistej. Rozważając eksperyment rzutu kostką nie jest dla nas istotne

co działo się z kostką, istotne jest jedynie ile oczek wypadło. Dlatego naturalnym wyborem zmiennej losowej jest funkcja określająca liczbę oczek które wypadły w wyniku rzutu kostką.

Typy zmiennych losowych:

- **ciągła** - zmienna przyjmuje dowolne wartości z określonego przedziału (w szczególności cały zbiór liczb rzeczywistych),
- **skokowa (dyskretna)** - zmienna przyjmuje dowolne wartości ze zbioru przeliczalnego (np. zbiór liczb całkowitych z określonego przedziału),
- **osobliwa** - zmienna losowa, której rozkład skupiony jest na nieprzeliczalnym zbiorze o długości 0 (np. na zbiorze Cantora), tzn. prawdopodobieństwo tego, że zmienna ta przyjmuje wartość z tego zbioru, wynosi 1, przy czym $P(X = x) = 0$ dla każdego $x \in \mathbb{R}$,
- dowolna zmienna losowa albo jest jednego z trzech powyższych typów, albo ma rozkład **mieszany** składający się z rozkładów tych typów.

9.4.2 Dystrybuanta i jej własności

Dystrybuanta zmiennej losowej X jest to funkcja zdefiniowana następująco:

$$F(x) = P(X \leq x). \quad (9.8)$$

(czytamy: „Dystrybuanta dla konkretnej wartości zmiennej losowej tj. dla $X=x$) jest równa prawdopodobieństwu tego, że zmienna losowa X będzie przyjmowała wartości nie większe niż konkretna wartość x ”.)

Własności dystrybuanty:

1. $0 \leq F(x) \leq 1$ dla każdego $x \in \mathbb{R}$,
2. $F(x)$ jest niemalejąca,
3. jest lewostronnie ciągła,
4. $\lim_{x \rightarrow -\infty} F(x) = 0$, $\lim_{x \rightarrow +\infty} F(x) = 1$.

Dystrybuanta zmiennej losowej *ciągłej* X jest to funkcja podana wzorem:

$$F(x) = P(X < x) = \int_{-\infty}^x f(x) dx. \quad (9.9)$$

Dystrybuanta zmiennej losowej *skokowej* X jest to funkcja podana wzorem:

$$F(x) = \sum_{x_i < p_i} p_i. \quad (9.10)$$

9.4.3 Funkcja gęstości

Funkcja gęstości zmiennej losowej ciągłej to funkcja $f(x)$ określona na zbiorze liczb rzeczywistych i spełniająca następujące warunki:

1. $f(x) \geq 0$ - jest określona nieujemnie,
2. $\int_{-\infty}^{+\infty} f(x) dx = 1$.

Mediana M to taka wartość zmiennej losowej x dla której dystrybuanta wynosi $1/2$.

$$F(x_{0.5}) = P(x < x_{0.5}) = 0.5 \quad (9.11)$$

Gęstość prawdopodobieństwa to szansa przyjęcia konkretnej wartości przez zmienną losową. Dystrybuanta to szansa przyjęcia przez zmienną losową wartości nie większej od argumentu.

9.4.4 Podstawowe parametry rozkładu zmiennej losowej

Wartość oczekiwana $E(X)$ (nadzieja matematyczna)

$$E(x) = m. \quad (9.12)$$

Wartość m jest to taka wartość zmiennej losowej X , wokół której skupiają się wyniki wielokrotnych realizacji tej zmiennej. Innymi słowy, oczekuje się (ma się nadzieję), że wielokrotne realizacje zmiennej losowej X będą skupiały się wokół liczby m . Wartość oczekiwana należy do miar położenia.

Dla zmiennych losowych **ciągłych** wzór wygląda następująco:

$$E(x) = \int_{-\infty}^{+\infty} x f(x) dx. \quad (9.13)$$

Dla zmiennych losowych **dyskretnych** wzór wygląda następująco:

$$E(x) = \sum_{i=1}^n p_i \cdot x_i. \quad (9.14)$$

Wariancja należy do miar rozproszenia. Można ją wyliczyć wykorzystując wzór:

$$V(x) = E[X - E(X)]^2 = E(X^2) - E(X)^2 \quad (9.15)$$

Odchylenie standardowe informuje jak szeroko wartości jakiejś wielkości są rozrzucone wokół jej średniej.

$$\sigma = \sqrt{V(x)}. \quad (9.16)$$

9.5 Ciągła, dyskretna i szybka transformata Fouriera, widmo sygnału

Transformata Fouriera jest wynikiem operacji zwanej transformacją Fouriera.

9.5.1 Ciągła transformata Fouriera

Jeżeli każdy skończony przedział $\langle a, b \rangle$ można podzielić na skończoną liczbę podprzedziałów, w których $f(x)$ jest monotoniczna oraz w każdym punkcie przedziału (a, b) są spełnione warunki Dirichleta i funkcja $f(t)$ jest całkowalna w przedziale $(-\infty, \infty)$, to funkcję:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt. \quad (9.17)$$

Transformacja Fouriera jest operacją odwracalną, zatem posiadając transformatę $\hat{f}(\omega)$ możemy wyznaczyć jej oryginał:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega t} d\omega. \quad (9.18)$$

9.5.2 Dyskretna transformata Fouriera

Dyskretna transformata Fouriera DFT:

$$A_k = \sum_{n=0}^{N-1} a_n w_N^{-kn}, \quad 0 \leq k \leq N-1 \quad (9.19)$$

$$w_N = e^{i \frac{2\pi}{N}} \quad (9.20)$$

Odwrotna DFT:

$$a_n = \frac{1}{N} \sum_{k=0}^{N-1} A_k w_N^{kn}, \quad 0 \leq n \leq N-1 \quad (9.21)$$

9.5.3 Szybka transformata Fouriera

Szybka transformacja Fouriera FFT:

9.5.4 Widmo sygnału

Widmem częstotliwościowym sygnału nazywa się przedstawienie sygnału w dziedzinie częstotliwości lub pulsacji otrzymane w wyniku transformacji Fouriera. Z wykresu widma możliwe jest odczytanie składowych harmonicznych sygnału. Charakterystyka widmowa istnieje tylko wtedy, gdy układ jest BIBO stabilny. Podstawienia $s = j\omega$ wolno dokonać tylko wtedy, gdy funkcja $H(s)$ nie ma biegunów w prawej domkniętej półpłaszczyźnie zmiennej s .

$A(\omega) = |H(j\omega)|$ - charakterystyka amplitudowa. Określa w jaki sposób modyfikowane jest widmo amplitudowe pobudzenia.

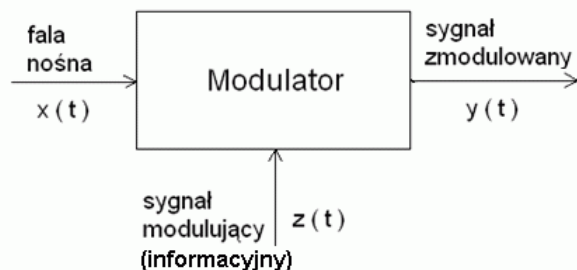
$\Theta(\omega) = \arg H(j\omega)$ - charakterystyka fazowa. Określa w jaki sposób jest modyfikowane widmo fazowe pobudzenia.

9.6 Modulacje analogowe i cyfrowe

Modulacja - celowa lub samorzutna zmiana parametrów sygnału.

Na wyjściu źródła informacji sygnał oryginalny jest przeważnie dolnopaśmowy. Jednak kanały radiowe i telekomunikacyjne są zawsze środkowoprzepustowe. Aby zamienić sygnał oryginalny na sygnał środkowopasmowy musimy zastosować metodę zwaną modulacją. Modulacja jest to uzmiennienie standardowego przebiegu $x(t)$ nazywanego sygnałem lub falą nośną przez sygnał $z(t)$ zwany sygnałem modulującym. W wyniku tego jako sygnał wyjściowy uzyskujemy sygnał zmodulowany $y(t)$. Schemat blokowy procesu modulacji został przedstawiony na rysunku 9.11. Inaczej mówiąc modulacja to zmiana parametrów fali nośnej, która może być np. sinusoidalna lub prostokątna. W przypadku modulowania fal sinusoidalnych zmianom może ulegać amplituda, częstotliwość lub faza. Natomiast dla fal prostokątnych szerokość, amplituda lub ilość impulsów. W zależności od charakteru zmian parametru modulowanego rozróżniamy modulacje analogowe (zmiany o charakterze ciągłym) oraz cyfrowe (zmiany o charakterze skokowym).

9.6.1 Modulacje analogowe



Rysunek 9.11: Schemat blokowy układu modulacji

Modulacja amplitudy AM (ang. Amplitude Modulation) ma miejsce, gdy w procesie modulacji następują zmiany amplitudy sygnału nośnego proporcjonalne do chwilowej wartości sygnału modulującego. Uzyskujemy ją poprzez zmianę parametru sygnału o częstotliwości nośnej. W tym przypadku jest to

amplituda sygnału nośnego Y_0 , która zmienia się pod wpływem sygnału modulującego $z(t)$. Uzyskujemy w ten sposób sygnał zmodulowany nadający się np. do transmisji drogą radiową. Proces modulacji jest realizowany w urządzeniu zwanym modulatorem. Modulację amplitudy AM charakteryzuje współczynnik głębokości modulacji m , zwany często głębokością modulacji. Definiujemy go jako stosunek amplitudy przebiegu modulującego M do amplitudy sygnału nośnego. Podawany jest on zazwyczaj w procentach.

$$m = M/Y_0. \quad (9.22)$$

Modulacja częstotliwości FM (ang. Frequency Modulation) ma miejsce, gdy w procesie modulacji zmianie ulega częstotliwość sygnału nośnego $x(t)$ pod wpływem sygnału modulującego $z(t)$. Częstotliwość chwilowa zmienia się według wzoru:

$$f(t) = f_0 + k \cdot z(t), \quad (9.23)$$

gdzie :

$z(t)$ - przebieg modulujący,

k - stała proporcjonalności.

Częstotliwość sygnału nośnego f_n zmienia się w pewnym zakresie, a różnica pomiędzy najniższą i najwyższą chwilową wartością częstotliwości fali nośnej nazywana jest dewiacją częstotliwości Δf . Stosunek dewiacji częstotliwości do częstotliwości sygnału nośnego:

$$m_f = \frac{\Delta f}{f_n}. \quad (9.24)$$

nazywamy współczynnikiem modulacji częstotliwości lub wskaźnikiem dewiacji częstotliwości. W zależności od wartości wskaźnika rozróżniamy szeroko-pasmową modulację częstotliwości (wskaźnik większy od jedności) oraz wąsko-pasmową modulację częstotliwości (wskaźnik mniejszy od jedności). Modulacja częstotliwości FM jest stosowana nie tylko do przesyłania sygnału radiowego w zakresie fal ultrakrótkich, ale również do transmisji sygnału w telewizji satelitarnej i sygnału dźwiękowego w wielu systemach telewizji naziemnej. Modulacja częstotliwości FM jest bardzo podobna do modulacji fazy PM. Gdy zmieniamy częstotliwość sygnału nośnego to zmianie ulega również faza. Identycznie odbywa się to w drugą stronę, gdy zmieniamy fazę sygnału nośnego, zmianie ulega również jego częstotliwość. Jednak modulacje FM i PM nie są sobie równoważne. Gdy odbiornik FM jest używany do demodulacji sygnału PM sygnał audio jest zniekształcony. Dzieje się tak dlatego, gdyż związek pomiędzy częstotliwością a fazą jest nieliniowy.

Modulacja fazy PM (ang. Phase Modulation) ma miejsce, gdy w procesie modulacji zmianie ulega chwilowa wartość fazy sygnału nośnego $x(t)$, która

zmienia się zgodnie z zależnością:

$$\Theta(t) = \omega t + k \cdot z(t), \quad (9.25)$$

gdzie:

$z(t)$ - przebieg modulujący,

k - stała proporcjonalności.

Modulacja częstotliwości pozwala na stosowanie prostszych modulatorów i demodulatorów sygnału. Dlatego też modulację fazy stosujemy głównie do transmisji cyfrowej. Charakterystyczny dla modulacji fazy jest współczynnik zwany dewiacją fazy, który definiujemy w następujący sposób:

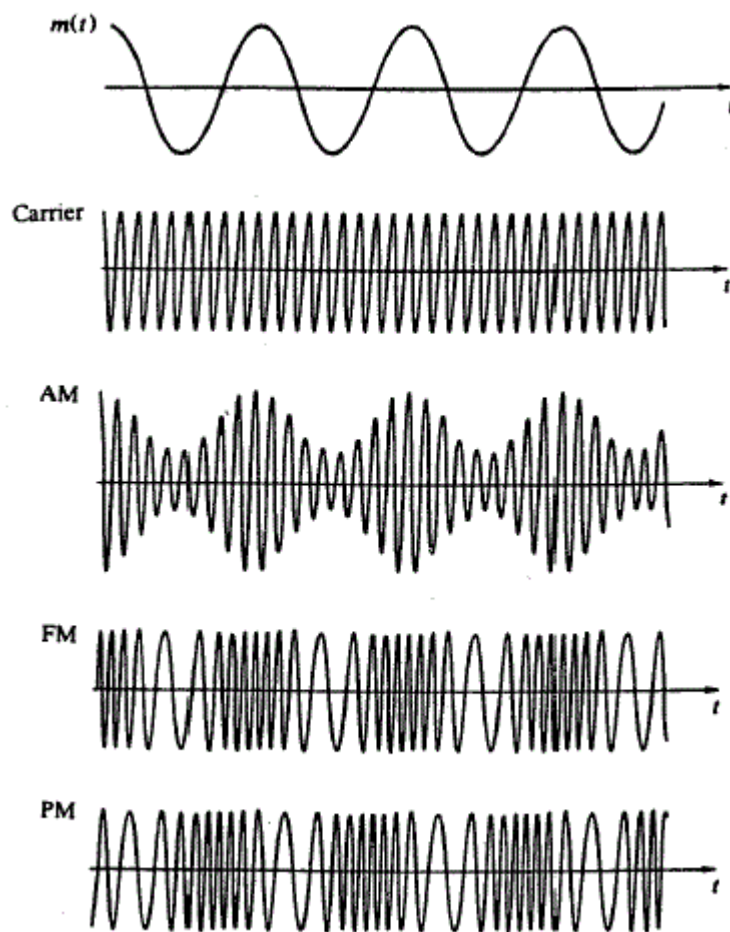
$$\Delta\phi_{PM} = k \cdot z_{max}. \quad (9.26)$$

9.6.2 Modulacje cyfrowe

Modulacja cyfrowa to proces zmiany analogowego sygnału nośnego przez binarny sygnał modulujący, który z łatwością możemy przesłać np. drogą radiową. Sygnałem nośnym w modulacji cyfrowej jak i analogowej jest przebieg sinusoidalny, jednak w modulacjach cyfrowych sygnał modulujący to strumień elementów binarnych. Podobnie jak w modulacjach analogowych tak i w cyfrowych zmianom ulega amplituda, częstotliwość lub faza przebiegu nośnego. Ponieważ cyfrowy sygnał modulujący zmienia się skokowo, to zmiany w przebiegu zmodulowanym również są skokowe. Celem modulacji jest dopasowanie właściwości sygnału wyjściowego zmodulowanego do parametrów kanału transmisyjnego.

Modulacja ASK (ang. Amplitude Shift Keying) zwana kluczkowaniem amplitudy ma miejsce, gdy występuje zmiana amplitudy sygnału nośnego w zależności od cyfrowego sygnału modulującego. Proces kluczkowania amplitudy jest zbliżony do modulacji amplitudy AM, a w zasadzie jest to szczególny rodzaj modulacji AM. Różnica polega na tym, iż sygnał modulujący jest sygnałem cyfrowym. Cechą charakterystyczną kluczkowania amplitudy jest to, że dzięki cyfrowemu sygnałowi modulującemu w czasie trwania stanu wysokiego występuje pełna amplituda sygnału zmodulowanego natomiast w stanie niskim jest ona wytłumiona. Podobnie jak modulacja AM, kluczkowanie amplitudy jest liniowe, czułe na zakłócenia atmosferyczne i zniekształcenia.

Modulacja FSK (ang. Frequency Shift Keying), zwana kluczkowaniem częstotliwości, jest szczególnym przypadkiem modulacji częstotliwości FM. Sygnałem modulującym jest sygnał cyfrowy. Modulacja FSK polega na przypisaniu odpowiedniej częstotliwości sygnału nośnego każdemu z dwóch stanów sygnału modulującego. Przejście z jednej częstotliwości może odbywać się z ciągłością fazy lub bez niej. Ta modulacja wyróżnia się stałą amplitudą chwilową, niezależną od sygnału modulującego. Ta właściwość jest bardzo przydatna przy



Rysunek 9.12: Modulacje

transmisji sygnałów przez kanały, gdzie występują zmiany amplitudy transmitowanego sygnału na skutek nieliniowości tego kanału. Zaletą tej modulacji jest również odporność na zakłócenia impulsowe oraz zniekształcenia tłumieniowe i opóźnieniowe. Dlatego też modulacja FSK jest częściej stosowana niż modulacja ASK.

Modulacja PSK (ang. Phase Shift Keying) zwana kluczowaniem fazy to jeden z rodzajów modulacji PM. Ma ona miejsce, gdy przy stałej amplitudzie i częstotliwości harmonicznego sygnału nośnego występuje przesunięcie fazy w zależności od stanu informacji pierwotnej. Sygnały przy modulacji PSK są zawsze transmitowane w systemach koherentnych, czyli faza sygnału nadawanego znana jest po stronie odbiorczej. Podobnie jak w przypadku modulacji FSK, modulacja PSK charakteryzuje się stałą w czasie amplitudą chwilową, co odróżnia ją od modulacji ASK, gdzie amplituda chwilowa zmienia się. Powoduje to większe narażenia na zniekształcenia nieliniowe.

Modulacja QPSK (ang. Quadrature Phase Shift Keying) zwana modulacją kwadraturową to jeden z rodzajów modulacji fazy. Może być ona traktowana jako klasyczna modulacja 4 - wartościowa PSK nośnej o amplitudzie A , bądź jako złożenie dwóch dwuwartościowych modulacji amplitudy BASK (ang. Binary Amplitude Shift Keying) o amplitudzie $A/\sqrt{2}$ i ortogonalnych nośnych $\sin 2\pi f_0 t$ oraz $\cos 2\pi f_0 t$. Zastosowano w niej kodowanie na czterech ortogonalnych przesunięciach fazowych sygnału nośnego. Jej zaletą jest zwiększenie efektywności wykorzystania pasma, przy jednoczesnym braku negatywnego wpływu na bitową stopę błędów. Modulację QPSK definiujemy:

$$x = A \cos(2\pi f t + \Theta), \quad 0 < t < T, \quad i = 1, 2, 3, 4 \quad \text{oraz} \quad \Theta = \frac{\pi(2i - 1)}{4} \quad (9.27)$$

Wyjściowe fazy sygnału to $\pi/4, 3\pi/4, 5\pi/4, 7\pi/4$. Częstotliwość nośnej jest wybierana jako całkowita wielokrotność szybkości nadawania znaku, dlatego faza sygnału jest jedną z wymienionych powyżej faz.

Modulacja MSK (ang. Minimum Shift Keying) jest szczególnym przypadkiem modulacji FSK, gdy częstotliwości f_1 i f_2 są równe odpowiednio $f_0 - 1/(4T_b)$ i $f_0 + 1/(4T_b)$. MSK może być również traktowana jako zmodyfikowana forma modulacji OQPSK (ang. Offset Quadrature Phase Shift Keying), w której elementy odpowiadające składowej synfazowej i kwadraturowej są odpowiednio ukształtowane, a następnie wymnożone przez przebiegi nośne. Przez ukształtowanie rozumiemy zastąpienie impulsu prostokątnego impulsem sinusoidalnym. Modulację MSK definiujemy jako:

$$s(t) = s_I(t) \cos(2\pi f_0 t) + s_Q(t) \sin(2\pi f_0 t). \quad (9.28)$$

Modulacja MSK zalicza się do klasy modulacji z ciągłą fazą CPM (ang. Continuous Phase Modulation).

Zastosowania:

- transmisja danych binarnych w kanale o wąskim paśmie,
- łączność modemowa, faksowa,
- łączność radiowa (telemetria, zdalne sterowanie),
- systemy bezprzewodowe,
- telefonia cyfrowa (GSM, UMTS, TETRA, ...),
- łączność satelitarna.

9.7 Wzmacniacze operacyjne: właściwości i zastosowania

Materiały: wzmacniacze_operacyjne.pdf

9.8 Mikroprocesory: budowa, zastosowania

9.8.1 Architektury

Architektura harwardzka W tej architekturze pamięć programu przechowująca rozkazy została oddzielona od pamięci danych (dwie różne magistrale). W trakcie pobierania argumentów wykonywanej właśnie instrukcji można równocześnie zacząć pobieranie następnego słowa rozkazowego (pre-fetch).

Architektura von Neumanna Architektura zakłada, że podział przestrzeni adresowej na pamięć programu i pamięć danych jest czysto umowny. W ten sposób jest uproszczony dostęp do obu pamięci, gdyż wykonywany jest za pomocą tych samych instrukcji. Główną wadą jest wydłużenie wykonywania cyklu.

Architektura super-harwardzka Pamięć programu i pamięć danych są oddzielone od siebie, ale wykorzystują one wspólną magistralę: danych i adresową.

CISC (ang. Complex Instruction Set Computing) to architektura konwencjonalnych procesorów. Charakteryzuje się ona znaczną liczbą elementarnych rozkazów i trybów adresowania przy niewielkiej liczbie rejestrów uniwersalnych. Jest ona bardzo wolna w porównaniu do RISC-ów. Na tej architekturze były oparte pierwsze procesory z rodziny x86. Dziś w tej rodzinie nie spotka się procesora CISC, ale jest ona zastąpiona przez wewnątrz RISC. Jednak pozostała zgodnie z zasadą "kompatybilności w tył" mała liczba rejestrów, co jest główną bolączką projektantów procesorów x86.

RISC (ang. Reduced Instruction Set Computing/Computer) wywodzi się z Berkeley (1985). Koncepcja tego typu architektury jest oparta na ograniczeniu liczby krótkich (maksymalnie dwusłowych) rozkazów mających niewiele formatów i trybów adresowania. Procesor RISC posiada wiele rejestrów uniwersalnych (nawet powyżej 100). Działanie procesora przyspieszają dodatkowe ulepszenia, np. przetwarzanie potokowe, czy pamięć podręczna. Instrukcje wykonują proste operacje, dzięki czemu mogą to robić szybko - każda jednostka wykonawcza jest w stanie wykonać dokładnie jedną instrukcję w jednym cyklu zegara. Lista instrukcji zawiera tylko kilka rozkazów umożliwiających dostęp do pamięci: załadowanie (load), zapis (store) oraz instrukcje semaforowe. Wszystkie pozostałe rozkazy operują wyłącznie na rejestrach - to również upraszcza budowę układu. Instrukcje kodowane są w prosty sposób - każdy rozkaz ma taką samą długość (np. 32 bity).

9.8.2 Budowa

Jeśli chodzi o budowę fizyczną, mikroprocesor to nic innego jak krzemowa płytką z milionem tranzystorów, które blokują lub umożliwiają przepływ prądu. Z tranzystorów budowane są bramki logiczne, a te z kolei są łączone w bardziej rozbudowane układy.

ALU (ang. Arithmetic Logic Unit) - wykonuje podstawowe operacje arytmetyczne (dodawanie, odejmowanie, dzielenie oraz mnożenie oraz logiczne (OR, AND, XOR, NOT) oraz przesunięcia bitowe. ALU współpracuje z roboczym rejestrem zwanym akumulatorem (lub wieloma akumulatorami), który przechowuje jeden z operandów (argumentów) wykonywanej operacji oraz wyniku tej operacji.

CU (ang. Control Unit) - dekoduje zawartość rejestru rozkazów i generuje odpowiednie sygnały sterujące zapewniające prawidłowy przebieg operacji zdefiniowanej kodem rozkazu.

Rejestry (ang. Register) - komórki pamięci do przechowywania tymczasowych wyników obliczeń, adresów lokacji w pamięci RAM itp. Rejestry są najszybszym rodzajem pamięci.

- **Rejestr instrukcji IR** (ang. Instruction Register) - przechowuje aktualnie wykonywaną instrukcję.
- **Licznik rozkazów PC** (ang. Program Counter) - przechowuje adres w pamięci, gdzie przechowywany jest kolejny rozkaz do pobrania. Rozkazy są przechowywane w postaci kodów binarnych.
- **Akumulator A** (ang. Accumulator) - przechowuje argument (operand) do operacji ALU lub wynik operacji.

- **Wskaźnik stosu SP** (ang. Stack Pointer) - wskazuje na szczyt stosu (adres ostatniej zapełnionej komórki stosu).
- **Rejestr flagowy** - przechowuje informacje dotyczące operacji ALU np. flaga przeniesienia lub pożyczki CF (ang. Carry Flag), flaga parzystości PF (ang. Parity Flag), flaga przepełnienia OF (ang. Overflow Flag) itp.

Magistrale (ang. Bus) - wewnętrzne szyny łączące.

- **szyna danych** (ang. data bus) - magistrala komunikacyjna wykorzystywana do przesyłania właściwych danych,
- **szyna adresowa** (ang. address bus) - łączy CPU z pamięcią. Określa pod jaki adres mają zostać wysłane dane szyną danych. Szerokość magistrali (liczba linii) określa maksymalną pojemność pamięci systemu (przestrzeń adresową)
- **szyna sterująca** (ang. control bus) - zapewnia regulację dostępu do szyny adresowej i szyny danych.

9.8.3 Mikroprocesor a mikrokontroler

Na system mikroprocesorowy składa się mikroprocesor, układy wejścia/wyjścia, pamięć programu i danych, szyny adresowe, szyny danych ..., system operacyjny. Mikrokontroler to pojedynczy układ scalony zawierający kompletny system, zdolny do samodzielnego wykonywania operacji arytmetycznych i logicznych oraz do sterowania układami i elementami zewnętrznymi. Typowy uC zawiera CPU, pamięć programu (często FLASH), pamięć danych RAM, układu wejścia/wyjścia, wewnętrzne źródło taktowania, interfejsy komunikacyjne oraz inne układy periferyjne np. kontroler przerwań, timery, ADC, DAC, DMA.

9.8.4 Zastosowania

- elektronika przemysłowa - sterowniki PLC,
- elektronika powszechnego użytku - telefony komórkowe, zegarki, komputery
- telekomunikacja - routery, switchy etc.,
- technika samochodowa - piloty, sterowniki świateł, lusterek, radia, kontrolery wtrysku,
- medycyna - ciśnieniomierze, EKG, USG, termometry, mierniki poziomu cukru we krwi,
- automatyka budynków - sterowniki klimatyzacji i rolet.

9.9 Sieci komputerowe: budowa, protokoły, zastosowanie

Sieć komputerowa - wzór wzajemnie połączonych komputerów, które mogą pracować samodzielnie i komunikować się z innymi komputerami.

9.9.1 Budowa

Składniki sieci komputerowych

- **hosty** - komputery wykorzystywane przez użytkowników,
- **serwery** - stale działające komputery o dużej mocy obliczeniowej świadczące usługi hostom (udostępnianie plików, baz danych itp.),
- **medium transmisyjne** - nośnik informacji (kable miedziane, światłowody lub fale radiowe),
- **sprzęt sieciowy** - koncentratory, przełączniki, routery, karty sieciowe, modemy, punkty dostępu,
- **oprogramowanie** - programy komputerowe zainstalowane na urządzeniach sieciowych.

Model odniesienia OSI (ang. Open System Interconnection)

- **warstwa aplikacji** - określa w jaki sposób aplikacje działają ze sobą,
- **warstwa prezentacji** - dodaje podstawowe formatowanie do prezentacji danych,
- **warstwa sesji** - zarządza przebiegiem komunikacji pomiędzy dwoma komputerami,
- **warstwa transportowa** - sprawdza poprawność wysyłanych danych,
- **warstwa sieci** - adresuje wiadomości wewnątrz i pomiędzy sieciami,
- **warstwa łącza danych** - określa sposób uzyskiwania dostępu do fizycznego medium,
- **warstwa fizyczna** - przesyła dane przez fizyczne medium.

Budowa (topologia) warunkowana jest przez zastosowanie sieci. Najprostsze komunikowanie się sieci można zrealizować przez jedynie połączenie komputerów, w innych przypadkach używa się urządzeń kierujących ruchem.

Topologia magistrali (szyny, linii) - połączone jednym, współdzielonym medium.

Zalety:

- Brak koncentratorów/przełączników
- Awaria węzła nie powoduje paraliżu sieci

Wady:

- Awaria kabla powoduje paraliż sieci
- Ograniczona możliwość rozbudowy
- Niska przepustowość
- Obsługuje tylko jeden kanał transmisyjny

Topologia gwiazdy - posiada punkt centralny (switch, koncentrator) i gwiazdździe połączone do niego komputery.

Zalety:

- Bardzo łatwa rozbudowa sieci
- Awaria węzła nie powoduje paraliżu sieci
- Wysoka przepustowość

Wady:

- Ograniczenie odległości stacji roboczej od koncentratora
- Uszkodzenie koncentratora powoduje całkowity paraliż sieci

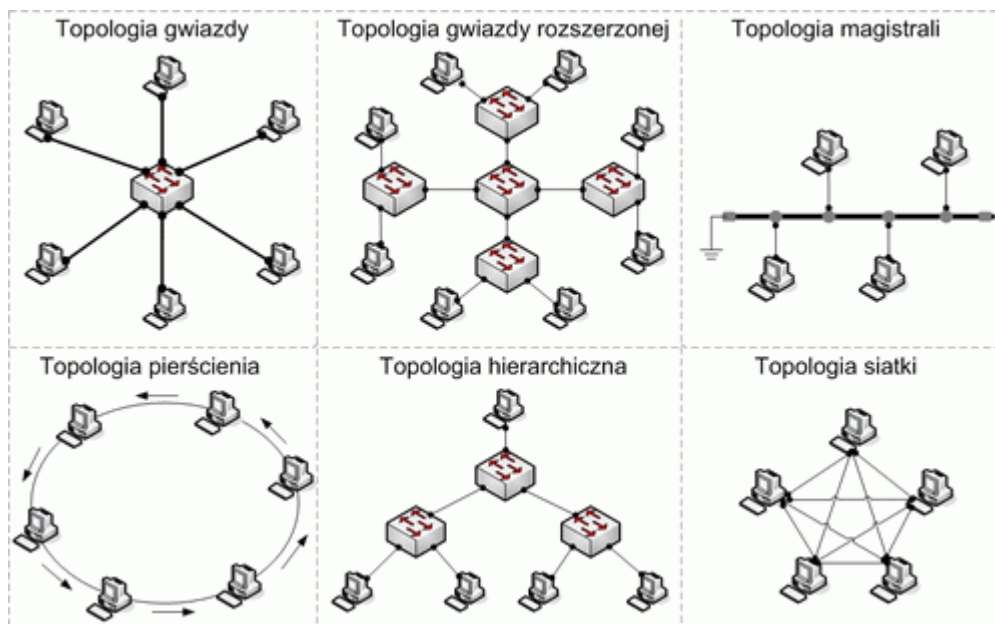
Topologia pierścienia - komputery połączone są za pomocą jednego nośnika informacji w układzie zamkniętym - okablowanie nie ma żadnych zakończeń (tworzy krąg).

Zalety:

- Niskie koszty budowy

Wady:

- Niska przepustowość
- Trudna do rozbudowy
- Ciężka lokalizacja uszkodzeń
- Uszkodzenie jednej stacji powoduje paraliż sieci



Rysunek 9.13: Topologie sieci komputerowych

Rozszerzone topologie

- **Topologia siatki**
- **Topologia gwiazdy rozszerzonej** – posiada punkt centralny (podobnie do topologii gwiazdy) i punkty poboczne (jedna z częstszych topologii fizycznych Ethernetu)
- **Topologia podwójnego pierścienia** – poszczególne elementy są połączone pomiędzy sobą odcinkami tworząc dwa zamknięte pierścienie
- **Topologia siatki** – oprócz koniecznych połączeń sieć zawiera połączenia nadmiarowe; rozwiązanie często stosowane w sieciach, w których wymagana jest bezawaryjność

Podstawowe urządzenia kierujące ruchem w sieci

- switch - urządzenie łączące segmenty sieci komputerowej pracujące głównie w drugiej warstwie modelu ISO/OSI (łącza danych), jego zadaniem jest przekazywanie ramki między segmentami sieci z doбором portu przełącznika, na który jest przekazywana.
- router - urządzenie sieciowe pracujące w trzeciej warstwie modelu OSI. Służy do łączenia różnych sieci komputerowych. Na podstawie informacji

zawartych w pakietach TCP/IP jest w stanie przekazać pakiety z dołączonej do siebie sieci źródłowej do docelowej, rozróżniając ją spośród wielu dołączonych do siebie sieci. Proces kierowania ruchem nosi nazwę trasowania, routingu lub routowania.

9.9.2 Protokoły

Protokół komunikacyjny - zbiór reguł i kroków postępowania, które są wykonywane podczas komunikacji co najmniej dwóch urządzeń ze sobą.

Protokoły warstwy aplikacji

FTP (ang. File Transfer Protocol) umożliwia dwukierunkowe przesyłanie plików binarnych i tekstowych, korzysta przy tym z protokołu TCP. FTP działa w oparciu o zasadę klient-serwer i korzystanie z usługi polega na użyciu interaktywnej aplikacji. Technologia FTP zapewnia ochronę stosując hasła dostępu.

HTTP (ang. Hypertext Transfer Protocol) odpowiada za przesyłanie dokumentów hipertekstowych w sieci WWW, która jest najszybciej rozwijającą się i najczęściej używaną częścią internetu. Przy jego pomocy przebiega komunikacja między klientami i serwerami sieci Web.

HTTPS (ang. HyperText Transfer Protocol Secure) to szyfrowana wersja protokołu HTTP. Zamiast używać w komunikacji klient-serwer niezaszyfrowanego tekstu, szyfruje go za pomocą protokołu SSL. Zapobiega to przechwytywaniu i zmienianiu przesyłanych danych. HTTPS działa domyślnie na porcie nr 443 w protokole TCP.

DNS (ang. Transmission Control Protocol) jest systemem tłumaczenia internetowych nazw domenowych na adresy IP. System DNS jest rozproszoną bazą danych obsługiwaną przez wiele serwerów, z których każdy posiada tylko informacje o domenie, którą zarządza, oraz o adresie serwera nadrzędnego. Na najwyższym poziomie znajdują się tzw. główne serwery nazw (root level servers), które znajdują się w Stanach Zjednoczonych i podłączone są do szybkich sieci szkieletowych Internetu. Przechowują one adresy serwerów nazw dla domen najwyższego poziomu, np. .com, .edu, .org, oraz domen krajowych, np. .pl, .de, .uk. Adresy serwerów głównych muszą być znane każdemu innemu serwerowi nazw. Wewnątrz każdej domeny można tworzyć tzw. subdomeny, np. wewnątrz domeny .pl utworzono wiele domen regionalnych jak .waw.pl, .lodz.pl itp, oraz funkcjonalnych jak .com.pl, .gov.pl lub .org.pl, należących do firm, organizacji lub osób prywatnych.

SMTP (ang. Simple Mail Transfer Protocol) odpowiada za przesyłanie poczty elektronicznej pomiędzy komputerami pracującymi w sieci.

POP3 (ang. Post Office Protocol v 3) pozwalający na odbiór poczty elektronicznej ze zdalnego serwera do lokalnego komputera poprzez połączenie TCP/IP.

TELNET (ang. Terminal emulation) - umożliwia użytkownikowi zdalny dostęp do innego komputera: zalogowanie się hoście internetowym i wykonywanie poleceń. Wysyłane dane nie są szyfrowane. Aktualnie częściej wykorzystywany jest protokół SSH (ang. Secure SHell) z szyfrowaniem danych.

Protokoły warstwy transportowej

TCP (ang. Transmission Control Protocol) działa w warstwie transportowej w *trybie połączeniowym*. Gwarantuje dostarczenie danych do odbiorcy. Połączenia TCP są połączeniami wirtualnymi, rozpoznawanymi po adresach i portach urządzeń docelowych i źródłowych. Połączenia takie charakteryzują się możliwościami sterowania przepływem, potwierdzaniem odbioru, zachowywaniem kolejności danych, kontrolą błędów i przeprowadzaniem retransmisji. Odbiorca po odebraniu danych zobowiązany jest do przesłania do nadawcy potwierdzenia odebrania danych. Jeżeli potwierdzenie nie nadejdzie w określonym czasie, to nadawca wysyła dane ponownie. Segmenty TCP składają się z nagłówka i danych.

UDP (ang. User Datagram Protocol) działa w warstwie transportowej w *trybie bezpołączeniowym*. Protokół ten nie gwarantuje dostarczenia danych do odbiorcy. Jeżeli pakiet nie dotrze do odbiorcy, lub dotrze uszkodzony, UDP nie podejmie żadnych działań zmierzających do retransmisji danych, a zapewnienie niezawodności pozostawi warstwie wyższej. Protokół wykorzystywany jest do szybkiego przesyłania danych w niezawodnych sieciach.

Protokoły warstwy sieci

IP (ang. Internet Protocol) zapewnia usługę dostarczania pakietów danych z jednego punktu sieci do drugiego. Nie analizuje zawartości pakietu, ale wyszukuje ścieżkę prowadzącą do jego miejsca przeznaczenia. Protokół ten wykorzystuje adresy sieciowe komputerów zwane adresami IP. Są to 32-bitowa liczba zapisywana jako sekwencje czterech ośmiobitowych liczb dziesiętnych (mogących przybierać wartość od 0 do 255), oddzielonych od siebie kropkami. Adres IP dzieli się na dwie części: identyfikator sieciowy (network id) i identyfikator komputera (host id). Istnieje kilka klas adresowych, o różnych długościach obydwu składników. Obowiązujący obecnie sposób adresowania ogranicza liczbę dostępnych adresów, co przy bardzo szybkim rozwoju Internetu jest dla niego istotnym zagrożeniem. W celu ułatwienia zapamiętania adresów wprowadzono nazwy symboliczne, które tłumaczone są na adresy liczbowe przez specjalne komputery w sieci, zwane serwerami DNS.

ARP (ang. Address Resolution Protocol) odpytuje wszystkie komputery w sieci, czy mają potrzebny mu adres IP i prosi o przesłanie odpowiadającego mu adresu fizycznego MAC. Aby ograniczyć ruch w sieci budowana jest dynamiczna tablica ARP, w której zapisywane są pary adres IP adres MAC komputerów z którymi został nawiązany kontakt. Tablica ta ma ograniczony rozmiar. Jeśli tablica ARP przepełni się, to jest z niej usuwany najstarszy wpis.

9.9.3 Zastosowania

- współdzielenie zasobów np. plików, drukarek,
- komunikacja np. poczta e-mail, telefonia,
- sieci przemysłowe,
- bezprzewodowe sieci czujników,