



Zajebiste odpowiedzi typie makaki są sztos

Sieć Aptek

i inne mały

June 25, 2022

Spis treści

I	Pytania - dr. hab. Bogdan Księżopolski	3
1	Protokoły TCP i UDP - porównanie i zastosowanie.	3
2	Protokół IP.	5
3	Atrybuty bezpieczeństwa informacji.	8
4	Składnia podstawowych zapytań języka SQL.	9
5	Metodyki zwinne (agile).	11

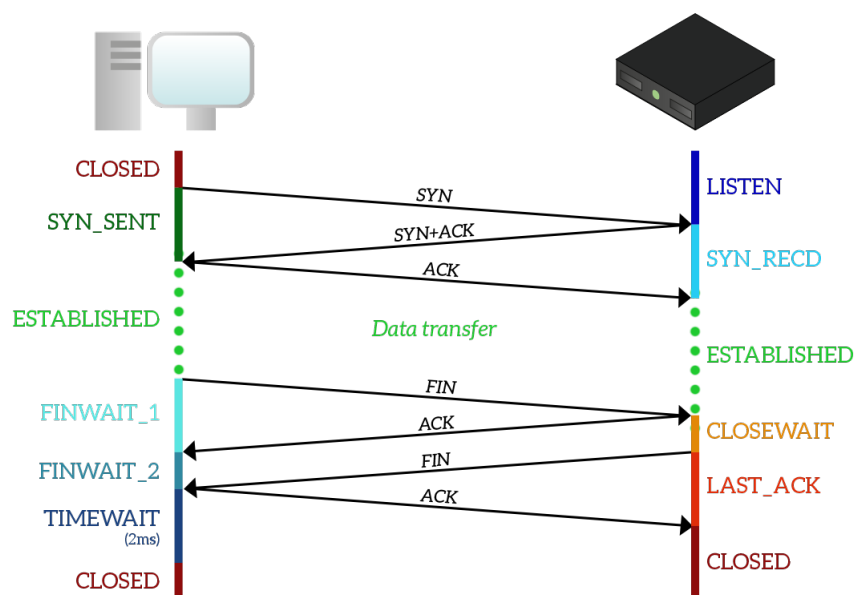
Rozdział I

Pytania - dr. hab. Bogdan Księżopolski

1 Protokoły TCP i UDP - porównanie i zastosowanie.

TCP

Protokół TCP lub Transmission Control Protocol jest protokołem zorientowanym na połączenie, znajdującym się w warstwie transportowej modelu TCP / IP. Nawiązuje połączenie między komputerem źródłowym a docelowym przed rozpoczęciem komunikacji.

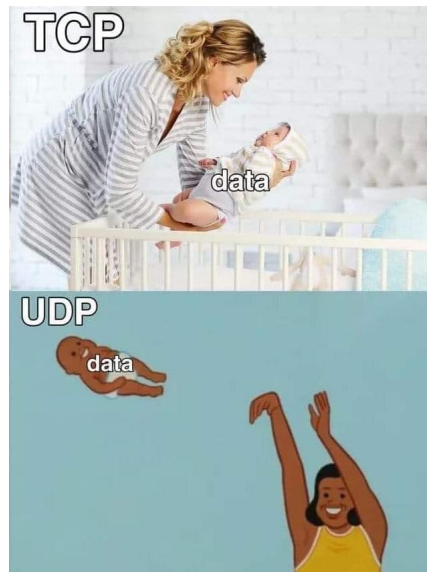


Rysunek I.1: TCP

Jest wysoce niezawodny, ponieważ wykorzystuje 3-drożną kontrolę uzgadniania, przepływu, błędów i przeciążenia. Zapewnia to, że dane wysyłane z komputera źródłowego są dokładnie odbierane przez komputer docelowy. Jeśli w przypadku, otrzymane dane nie są w odpowiednim formacie, to TCP ponownie przesyła dane. Poniższe protokoły używają TCP do transmisji danych:

- HTTP
- HTTPs
- FTP
- SMTP

UDP



Rysunek I.2: UDP :D

Protokół UDP lub User Datagram Protocol to bezpołączeniowy protokół znajdujący się w warstwie transportowej modelu TCP / IP. Nie ustanawia połączenia ani nie sprawdza, czy komputer docelowy jest gotowy do odbioru, czy też nie, po prostu przesyła dane bezpośrednio. Protokół UDP służy do przesyłania danych z większą szybkością. Jest mniej niezawodny i dlatego jest używany do przesyłania danych, takich jak pliki audio i wideo. UDP nie gwarantuje ani dostarczenia danych, ani nie przesyła utraconych pakietów.

2 Protokół IP.

IP (Internet Protocol)

Protokół internetowy jest protokołem komunikacyjnym warstwy Internet w modelu TCP/IP (odpowiada warstwie sieciowej modelu OSI). Protokół ten definiuje zasady i sposoby postępowania urządzeń sieciowych w celu nawiązania połączenia, utrzymania go i samej transmisji danych. Protokół IP stosowany jest w większości rodzajów sieci, w tym w sieci lokalnej i sieci Internet (każdy host, np. komputer, posiada swój własny, unikalny dla sieci adres IP).

Dane z użyciem protokołu IP transmitowane są w pakietach (paczках danych). Nie gwarantuje on jednak dotarcia danych do celu czy utrzymania kolejności pakietów. Może się zdarzyć, że odbiorca otrzyma kilkakrotnie ten sam pakiet z całej paczki danych, pakiety dotrą w innej kolejności lub nie dotrą w ogóle. W celu zapewnienia prawidłowej transmisji stosuje się różne techniki w wyższej warstwie, np. z użyciem protokołu TCP.

Ponieważ każdy host w sieci posiada swój własny unikalny adres IP, obecnie wykorzystywana czwarta wersja protokołu (v4) okazała się niewystarczająca i brakuje wolnych adresów IP. W tym celu utworzona została wersja szósta (v6) znacznie zwiększająca ilość różnych adresów IP. Same adresy IP dzielone są na kilka grup z których 3 najważniejsze to adresy publiczne, adresy prywatne (do wykorzystania w sieciach domowych, np. 192.168.1.1), oraz adresy pętli zwrotnej (np. 127.0.0.1).

W skrócie:

- protokół komunikacyjny z warstwy trzeciej (sieci)
- jest to protokół bezpołączeniowy
- głównym zadaniem tego protokołu jest przypisywanie każdemu urządzeniu sieciowemu adresu IP i wybór trasy w celu przesłania pakietów z danymi (w przypadku problemów w przesyłaniu pakietów protokół wybierze trasy alternatywne do przesłania pakietów)
- nie zapewnia dostarczania pakietów (nie posiada mechanizmów retransmisji, lecz na szczęście za to odpowiadają protokoły z warstw wyższych)

Klasy IP

Klasa	Zakres adresów pierwszego oktetu	Standardowa maska podsieci
A	0 – 127	255.0.0.0
B	128 – 191	255.255.0.0
C	192 – 223	255.255.255.0
D	224 – 239	–
E	240 – 255	–

Rysunek I.3: Klasy IP

Adresy klasy A przeznaczone są dla dużych sieci. Pierwszy bit oktetu w którym zawarty jest adres sieci jest równy 0. W związku z tym adresy sieci mogą przyjmować wartości od 0 do 127. Sieci 0 i 127 są zarezerwowane, więc do wykorzystania pozostają sieci od 1 do 126. W każdej sieci należącej do klasy A możemy wyodrębnić 16777216 adresów (liczba urządzeń będzie o 2 mniejsza, ale o tym w dalszej części artykułu). Klasa 127.0.0.0 wykorzystywana jest na potrzeby pętli zwrotnej, tj. umożliwia wysyłanie pakietów do samego siebie. Maska standardowa dla tej klasy to 255.0.0.0.

Adresy klasy B przeznaczone są do sieci średniej wielkości. Adres sieci zawarty jest w dwóch oktetach. Pierwsze dwa bity pierwszego oktetu wynoszą 10. W każdej sieci należącej do tego klasy można wyróżnić 65536 adresów (65534 urządzenia). Do tej klasy należą adresy sieci od 128 do 191 w ujęciu dziesiętnym. Maska standardowa dla tej klasy to 255.255.0.0.

Adresy klasy C przeznaczone są dla małych sieci, gdyż każda sieć może posiadać „jedynie” 256 adresów (254 urządzenia). Na adres sieci w sieciach należących do tej klasy przeznaczone są 3 oktety. Pierwsze trzy bity adresu wynoszą 110, w związku z tym do klasy tej należą adresy od 192 do 223 dziesiętne. Maska standardowa dla tej klasy to 255.255.255.0.

Klasa D została zarezerwowana na potrzeby rozsyłania grupowego przy użyciu adresów IP. Adres należący do tej klasy umożliwia przekierowanie pakietów do zdefiniowanej wcześniej grupy odbiorców. Dzięki temu możliwe jest przesłanie danych równocześnie do wielu odbiorców. Adresy tej klasy wykorzystywane są np. przez protokoły routingu. Pierwsze cztery bity adresu IP są równe 1110. Adresy należące do tej klasy zawierają się w przedziale od 224 do 239.

Adresy należące do klasy E zostały zarezerwowane przez Internet Engineering Task Force na potrzeby badawcze, wobec tego nie są dostępne publicznie. Pierwsze cztery bity adresu klasy E mają wartość 1111, w związku z tym adresy tej klasy zawierają się w przedziale od 240 do 255 dziesiętnie.

Prywatne adresy IP

Adresy prywatne wg klas:

- Klasa A – 10.0.0.0 – 10.255.255.255 z maską 255.0.0.0
- Klasa B – 172.16.0.0 – 172.31.255.255 z maską 255.255.0.0
- Klasa C – 192.168.0.0 – 192.168.255.255 z maską 255.255.255.0

Rodzaje trasowania protokołu IP

1. Anycast - dane są wysyłane do (topologicznie) najbliższego odbiorcy
2. Broadcast - dane wysyła do wszystkich możliwych hostów
3. Multicast - dane są wysyłane do wielu wybranych hostów (np. do hostów należących do jednej grupy)

4. Unicast - dane są wysyłane do jednego odbiorcy
5. Geocast - dane są wysyłane do wielu wybranych hostów należących do jednej strefy geograficznej

3 Atrybuty bezpieczeństwa informacji.

Triada bezpieczeństwa CIA - Confidentiality, Integrity, Availability

1. Poufność - osoby nieupoważnione nie mają dostępu do informacji podczas przechowywania, przetwarzania i przesyłania (np. zaszyfrowanej wiadomości)
2. Integralność - zapewnia że wiadomość nie została zmodyfikowana (w trakcie przechowywania, transportowania lub przetwarzania)
3. Dostępność - określa nieprzerwany dostęp do zasobów (w odpowiednio szybkim czasie) przez osoby aktualnie mające dostęp (np. osoby ze starej sesji nie mają już dostępu) i zapobieganie atakom DoS.
4. Niezaprzeczalność (nie wlicza się do triady) - podmiot nie może zaprzeczyć że wykonał jakąś czynność (np. że użytkownik wysłał wiadomość lub chciał uzyskać dostęp do strony poprzez logowanie).
5. Autentyczność - zapewnia że dane które zostały przyjęte pochodzą od osoby która faktycznie jest wysłała (jest zapewnione że nikt się nie podszył wysyłając te dane)
6. Rozliczalność - polega na rejestrowaniu działań/czynności wykonywanych w konkretnym czasie przez dane osoby/procesy (np. Janek logował się na strony +18 30 maja 2018 o 16:54:43)

4 Składnia podstawowych zapytań języka SQL.

DDL - Data Definition Language

Jak zapamiętać skrót:

Definition = DEFINIOWANIE - TWORZENIE STRUKTUR

W skład DDL wchodzi DROP, CREATE oraz ALTER.

- DROP - usunięcie struktury

```
DROP TABLE tabela;
```

- CREATE - stworzenie struktury

```
CREATE TABLE tabela(  
  kolumna1 typ (rozmiar),  
  kolumna2 typ (rozmiar),  
  ...  
);
```

- ALTER - modyfikacja struktury. Obejmuje operacje takie jak np. dodanie kolumny do tabeli, zmiana typu danych w kolumnie, usunięcie kolumny.

```
ALTER TABLE table  
ADD kolumna typ(dlugosc);
```

DML - Data Manipulation Language

Jak zapamiętać skrót:

Manipulation = MANIPULACJA - EDYCJA LUB TWORZENIE REKORDÓW

W skład DML wchodzi INSERT, UPDATE oraz DELETE.

- INSERT - dodawanie wierszy

```
INSERT INTO tabela (kolumna1, kolumna2, ..., kolumna_n)  
VALUES (wartosc1, wartosc2, ..., wartosc_n)
```

Nie trzeba podawać kolumn po nazwie tabeli gdy podamy po pierwsze wszystkie wartości, a po drugie w dobrej kolejności

- UPDATE - aktualizowanie danych, zmiana

```
UPDATE tabela  
SET kolumna1 = wartosc, kolumna2 = wartosc2, ...  
WHERE warunek
```

WHERE nie jest konieczny, możemy go użyć jak chcemy doprecyzować które rekordy mają się zaktualizować

- DELETE - usuwanie wierszy

```
DELETE FROM tabela  
WHERE warunek
```

WHERE nie jest konieczny, możemy go użyć jak chcemy doprecyzować które rekordy mają się skasować

DCL - Data Control Language

Jak zapamiętać skrót:

Control = KONTROLA = UPRAWNIENIA

W skład DCL wchodzi GRANT, REVOKE oraz DENY.

- GRANT - nadawanie uprawnień do pojedynczych obiektów lub globalnie konkretnemu userowi
- REVOKE - odbieranie uprawnień konkretnemu userowi
- DENY - zabranianie wykonywania operacji
- Składnia jest taka sama dla w/w poleceń:

```
[GRANT/REVOKE/DENY] operacja1, operacja2, ...  
ON tabela  
TO user
```

Przykład:

```
GRANT SELECT, INSERT  
ON Fragment  
TO glazik
```

DQL - Data Query Language

Jak zapamiętać skrót:

Query = ZAPYTANIA = SELECTY

W skład DQL wchodzi jedno polecenie: SELECT. Pozwala wybierać wiersze z bazy danych. Składnia:

```
SELECT kolumny  
FROM tabele  
WHERE warunek  
GROUP BY kolumna  
HAVING warunek  
ORDER BY ... DESC/ASC;
```

Dodatkowe klauzule SELECT:

- ORDER_BY - sortowanie wyników względem np. kolumny
- ASC oraz DESC - dodawane po sortowaniu, wybieramy czy ma być ascending czy descending
- GROUP_BY - grupowanie wyników względem danej kolumny
- HAVING - filtrowanie grup, działa podobnie do WHERE. WHERE oraz HAVING mogą występować jednocześnie w zapytaniu.

5 Metodyki zwinne (agile).

Programowanie zwinne (agile) - są to metodyki wytwarzania oprogramowania które są oparte na programowaniu iteracyjno-przyrostowym oraz na obserwowaniu czy wymagania nie ewoluują.

Cechy:

- przeznaczony głównie dla małych zespołów programistycznych
- w zespołach nie występuje hierarchia
- zespoły same się organizują
- komunikacja jest jednym z głównych elementów podczas produkcji oprogramowania
- szybkie wytwarzanie oprogramowania (i dobrej jakości)
- oprogramowanie jest dostarczane cyklicznie
- mała ilość dokumentacji

Manifest Agile (ważniejsze i mniej ważne [ale występuje]):

- Ludzie i interakcje i Procesy i narzędzia
- Działające oprogramowanie i Obszerna dokumentacja
- Współpraca z klientem i Formalne ustalenia
- Reagowanie na zmiany i Podążanie za planem

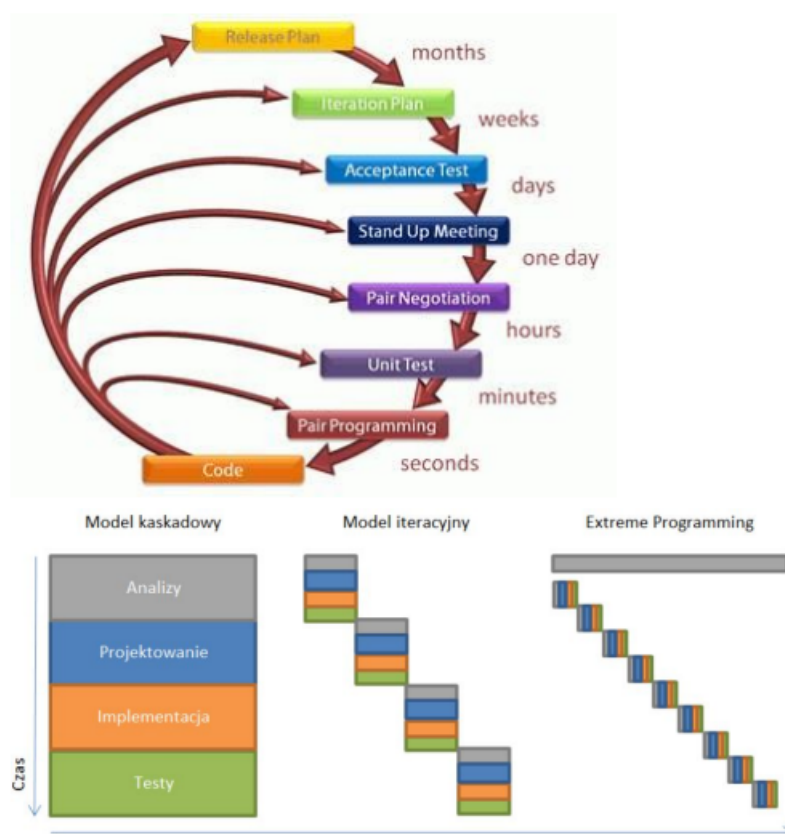
Etapy Agile:

1. Planowanie - zbieranie wymagań klienta i ich analiza (kluczowa jest umiejętność dobrej komunikacji z klientem)
2. Projektowanie - polega na przemyśleniu jak wykonać zaplanowany element
3. Programowanie
4. Testowanie
5. Informacja zwrotna - zgłaszanie niewykrytych błędów, nowych pomysłów i zmian wymagań klienta

Metodyki zwinne

1. XP (eXtreme Programming)
 - stosowany w małych i średnich projektach “wysokiego ryzyka” (czyli takich gdzie nie wiadomo do końca jak zrealizować dany cel)
 - tutaj najpierw są pisane testy, a potem program
 - na początku projekt ma za zadanie spełniać minimalne wymagania lecz potem jest rozwijany i z każdą iteracją (co kilka tygodni) następuje kontakt z klientem który wyraża swoją opinię (informację zwrotną)

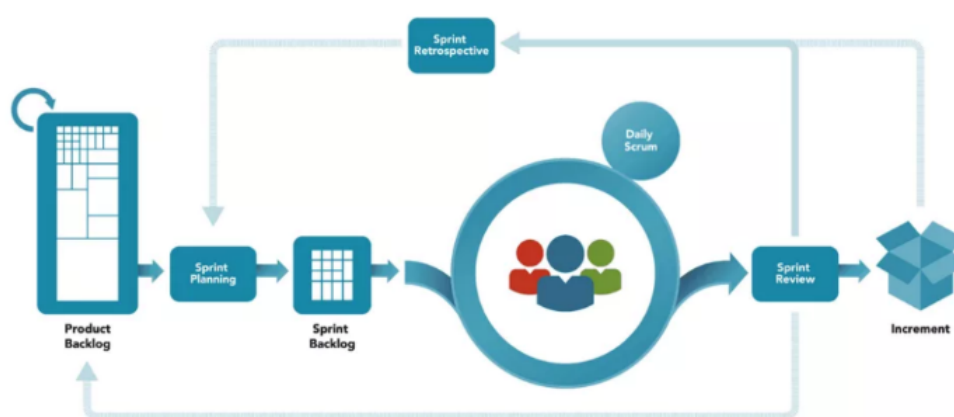
- występuje kontakt klient-programista
- podstawą jest dobra komunikacja ustna w zespole (brak dokumentacji)
- programiści pracują parami (jeden pisze, a drugi poprawia i komentuje; po jakimś czasie następuje zamiana)
- kod jest wspólny dla wszystkich i każdy może coś zmienić
- brak dokładnej specyfikacji wymagań
- iteracja posiada 4 etapy (wykonywane równoległe, a nie po sobie) : Planowanie, Projektowanie, Programowanie, Testowanie



2. Scrum

- iteracje wytwarzania oprogramowania są nazywane Sprintami (iteracja trwa najwyżej miesiąc)
- każdy sprint dodaje nową funkcjonalność (jak w większości metodyk)
- zespół się sam organizuje (podobnie jak w XP)
- występują tutaj codzienne 15 minutowe spotkania całego zespołu w celu omówienia projektu (co kto zrobił, co zamierza zrobić)
- Podczas produkcji występują 3 role:
 - Development Team - to grupa od 3 do 9 osób która tworzy produkt
 - Product Owner - osoba reprezentująca klienta (bierze ciągły udział podczas produkcji)

- Scrum Master - osoba która wspiera zespół, pomaga gdy zespół o to poprosi, interweniuje gdy trzeba, pracuje z Product Ownerem (to taki opiekun)
- proces wytwórczy:
 - zbieranie listy wymagań i określanie cech systemu (czyli funkcjonalności istotnych z punktu widzenia klienta)
 - Product Owner przedstawia priorytety i główny cel (z tych wymagań)
 - (początek sprintu) wybieranie zadań o najwyższym priorytecie
 - tworzenie oprogramowania (programiści sami wybierają które zadanie chcą robić [no z tych priorytetowych oczywiście]) z udziałem Product Ownera i występujące codzienne 15 minutowe spotkania
 - (koniec sprintu) prezentowanie wyników ze sprintu



3. TDD (Test-driven development)

- polega on na tym że na początku są pisane testy, a potem kod
- przydatny dla większych projektów (dla małych to jest strata czasu z pisanem tych testów)
- etapy cyklu tworzenia oprogramowania:
 - napisanie testów (które nie przechodzą przy braku implementacji)
 - napisanie kodu które przejdzie test
 - refaktoryzacja kodu
- pisanie testów na początku zmniejsza ryzyko występowania błędów
- to jest technika “pierwsze testy, a potem kod” zmniejszający ryzyko występowania błędu, a gdy jest odwrotnie to nie dość że trzeba szukać błędu to i ilość kodu jest większa (od powstania błędu)

