**Prednáška 8**

Ciele učenia

* Dôvody pre špecifikovanie prezentačnej vrstvy
* Funkcie prezentačnej vrstvy
  + Konverzia dát
  + Komprimovanie dát
  + Šifrovanie dát
* Princípy funkcií prezentačnej vrstvy
* Protokoly prezentačnej vrstvy

Prezenčná vrstva

OSI model

Prezentačná vrstva

Charakteristika

* je šiesta vrstva OSI modelu
* poskytuje služby aplikačnej vrstve
* definuje spôsob, akým aplikácia komunikuje so sieťou
* zabezpečuje, aby vyslané dáta boli interpretované rovnako u prijímateľa
* poskytuje ďalšie služby komfortu prenosu

Prečo potrebujeme prezentačnú vrstvu?

* Používateľom sa môže zdať, že stačí, ak dokážu vzájomne komunikovať dva programy na aplikačnej vrstve
* K tomu, aby mohli komunikovať potrebujú rovnakú interpretáciu vymieňaných dát
* Tá nie je vždy zaručená aplikáciou

Príklady rôznej interpretácie

* Niektoré počítače firmy IBM používajú pre kódovanie znakov kód EBCDIC, zatiaľ čo prevažná väčšina ostatných používa k rovnakému účelu kód ASCII.
* K znázorneniu celých čísiel so znamienkom používa väčšina počítačov dvojkový doplnkový kód, ale napríklad. počítače CDC Cyber pracujú s jednotkovým doplnkovým kódom.
* Mikroprocesory 80x86 firmy Intel číslujú jednotlivé bajty jedným smerom, zatiaľ čo mikroprocesory radu M68000 firmy Motorola číslujú jednotlivé bajty presne opačne.
* Veľmi časté sú odlišnosti napríklad vo formáte čísiel v pohyblivé rádovej čiarke, odlišné rozsahy zobraziteľných celých čísel, dané počtom k tomu vyhradených bitov
* ...

Prečo potrebujeme prezentačnú vrstvu?

* Reprezentácia celočíselných typov premenných na rôznych procesoroch
  + Procesory Intel sú tzv. little-endian: číselnú premennú ukladajú  
    v poradí od najnižšieho bajtu k najvyššiemu
  + Procesory PowerPC sú tzv. big-endian: číselnú premennú ukladajú v poradí od najvyššieho bajtu k najnižšiemu
  + Napr. hodnotu 0x0A0B (2571) si procesor Intel ukladá v pamäti ako 0x0B0A, PowerPC ako 0x0A0B
* Reprezentácia reťazcov v rôznych programovacích jazykoch
  + Pascal: Prvý bajt uvádza dĺžku reťazca, reťazec je za ním
  + C: Reťazec je ukončený bajtom s hodnotou 0

Prečo potrebujeme prezentačnú vrstvu?

* Rôzne kódové stránky pre zápis znakov národnej abecedy (diakritika)
  + Tradičné kódovanie je ASCII, ktoré však neobsahuje napr. slovenské diakritické znamienka
  + Kódovania slovenskej diakritiky sú napr. ISO8859-2, Kamenický, Windows-1250, UTF-8...
  + Problém je markantný najmä na webových stránkch
* Rôzne formáty prenášaných dát rovnakého typu
  + Obrázok? Aký formát? JPG, BMP, PNG, ...
  + Video? Aký formát? MPEG, DivX, H.264, Theora, ...
  + Hlas? Aký formát? G.711, G.729, EFR, Speex...
  + Dáta? Aký formát? EXE, DOC, PDF, XLS, ...

Prezentačná vrstva

* Prezentačná vrstva odbremeňuje aplikáciu, aby sa nemusela zaoberať podrobnosťami o prenášaných dátach
* V TCP/IP architektúre sú funkcie prezentačnej vrstvy obsiahnuté v aplikáciách
  + V súvislosti s TCP/IP sa preto nezvykne hovoriť o prezentačných protokoloch
  + Napriek tomu existujú štandardizované mechanizmy, ako si má aplikácia jednotlivé funkcie prezentačnej vrstvy zariadiť

Funkcie prezentačnej vrstvy

* Základná funkcia je zaistenie rovnakého formátu dát na vysielacej a prijímacej strane
* Realizácia tejto funkcie sa rozčleňuje:
  + Identifikácia formátu
  + Konverzia formátu
  + Dohodnutie formátu
* Ďalšie funkcie:
  + Kompresia prenášaných dát
  + Šifrovanie a zabezpečenie prenášaných dát

Formát dát

* Formát dát znamená tvar v akom majú byť dáta vyjadrené
* Formát dát je závislý od typu informácie:
  + Text
  + Obraz
  + Grafika
  + Štruktúrované dáta
* Formát je rôzny aj pre rôzne aplikácie

Zabezpečenie rovnakého formátu

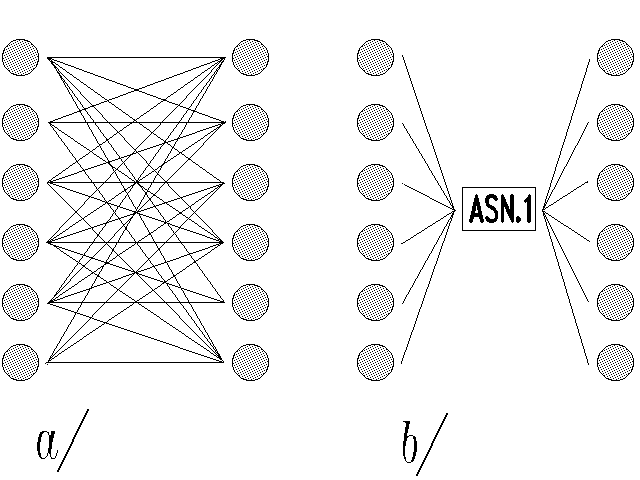
* Existuje mnoho možností, ako túto službu realizovať
* Najčastejšie používané princípy:
  + zvláštny jazyk dátových štruktúr pre prenos
    - Abstract Syntax Notation 1 (ASN.1)
  + identifikácia typu prenášanej správy
    - Multipurpose Internet Mail Extensions (MIME) rozšírený štandard internetu
    - Type-Length-Value (TLV)

Zabezpečenie rovnakého formátu

Všeobecne

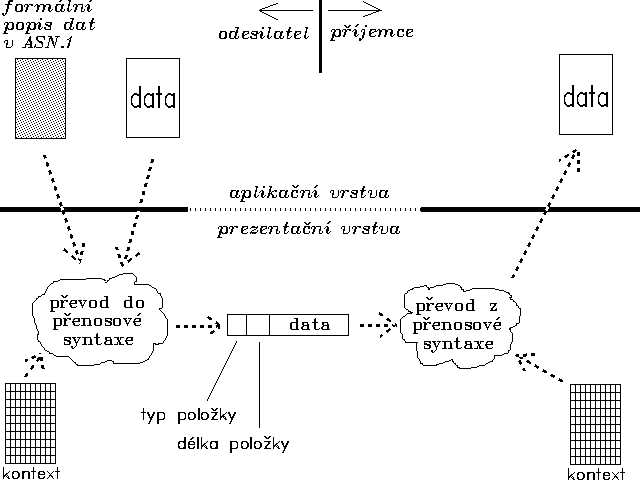
* Konverzia na úrovni prezentačnej vrstvy
* Dva typy konverzie

Každý s každým spoločný medzi tvar pre prenos



Jednotné zabezpečenie formátu

* Cez zvláštny formálny jazyk ASN.1 *(Abstract Syntax Notation)*



Abstract Syntax Notation 1

* ASN.1 je popisný jazyk, ktorý stanovuje „gramatiku“ celého protokolu
  + Princíp jazyka sa podobá XML
* V ASN.1 sú špecifikované celé formáty paketov týchto protokolov:
  + H.323
  + Simple Network Management Protocol
  + Lightweight Directory Access Protocol
  + DER formát digitálnych certifikátov
  + ... a mnohé ďalšie
* Príklad tvaru tohto jazyka:

FooProtocol DEFINITIONS ::= BEGIN

FooQuestion ::= SEQUENCE {

trackingNumber INTEGER,

question IA5String

}

FooAnswer ::= SEQUENCE {

questionNumber INTEGER,

answer BOOLEAN

}

END

myQuestion FooQuestion ::= {

trackingNumber 5,

question "Anybody there?"

}

Identifikovanie typu formátu

Protokol MIME - Multipurpose Internet Mail Extensions

* MIME pôvodne vzniklo ako rozšírenie služieb elektronickej pošty
  + Prenos správ v inej kódovej sade než ASCII
  + Prenos binárnych správ
  + Viac časťové telá správ (prílohy)
  + Informácie v hlavičke v inej kódovej sade než ASCII
* MIME je popísané v dokumentoch RFC 2045 – 2047 a niektorých ďalších (http://www.rfc-editor.org)
* MIME je však natoľko univerzálne navrhnuté, že sa používa aj v mnohých ďalších službách, napr. http

Multipurpose Internet Mail Extensions

* V MIME sa do prenášaných správ vkladá ako textové riadky v tvare

MENO: HODNOTA

* Najčastejšie je možné stretnúť sa s riadkom

Content-Type: typ-obsahu

kde typ-obsahu môže nadobúdať hodnoty napr.:

* + application/x-msdos-program
  + image/jpeg
  + text/html

Multipurpose Internet Mail Extensions

* Hlavička obyčajného e-mailu

Date: Thu, 10 Apr 2008 05:41:53 +0200

From: =?UTF-8?B?UGV0ZXIgUGFsw7pjaA==?= <Peter.Paluch@fri.uniza.sk>

MIME-Version: 1.0

To: =?UTF-8?B?VmxhZGltw61yIE1hcnR5xaHEjcOhaw==?= <vladimir.martyscak@ingn.sk>

Subject: Aktualizovany program?

Content-Type: text/plain; charset=UTF-8

Content-Transfer-Encoding: quoted-printable

* Hlavička odpovede HTTP servera na žiadosť o nejakú stránku

HTTP/1.0 200 OK

Server: Microsoft-IIS/5.0

Date: Thu, 10 Apr 2008 21:08:04 GMT

Content-Length: 64019

Content-Type: text/html; charset=Windows-1250

Connection: keep-alive

* Uvedené MIME prvky nie sú jediné, no sú najčastejšie používané
* MIME sa využíva v rôznych službách:
  + E-mail
  + HTTP
  + POP3, IMAP
  + SIP

Type-Length-Value – LTV

* MIME je ľahko čitateľný a zrozumiteľný formát vhodný najmä pre textové protokoly, pre binárne protokoly sa ale neveľmi hodí
* Tieto protokoly používajú iný prístup, nazývaný TLV
* TLV je principiálne veľmi jednoduchá idea
  + Možné typy správ si očíslujeme
  + Ku každej správe priložíme informáciu o jej type a dĺžke správy
  + Odosielané správy tvoria teda opakujúcu sa postupnosť <Typ, Dĺžka, Hodnota>, z čoho vyplýva aj názov tejto metódy

Type-Length-Value

* Príklad:
  + Typy správ: Meno, Priezvisko, E-mail
  + Očíslovanie: 1, 2, 3
* Postupnosť správ „Alfonz“, „Vrtinôžka“, „vrtinoh@laba.sk” by bola odoslaná ako

<1,6,Alfonz><2,9,Vrtinôžka><3,15,vrtinoh@laba.sk>

* TLV prístup má mnohé výhody
  + V jednom datagrame môže ísť za sebou ľubovoľný počet správ
  + Na poradí správ nezáleží
  + Ak príjemca nepozná typ konkrétnej správy, stačí mu obskočiť príslušný počet bajtov a pokračovať ďalšou správou
* TLV je využívané v mnohých protokoloch a mnohokrát aj na inej než len na prezentačnej vrstve
  + Cisco Discovery Protocol, Link Layer Discovery Protocol
  + Smerovací protokol IS-IS
  + ICQ – protokol OSCAR
  + RADIUS

Dojednanie spoločného formátu dát

* Ďalšou zo služieb prezentačnej vrstvy je vzájomné dohodnutie sa oboch komunikujúcich strán na spoločnom formáte dát
* Jedná sa o mierne inú službu než pri identifikovaní typu dát
  + Identifikovať znamená len určiť typ, ale neznamená to, že dané dáta vieme spracovať
  + Dojednať spoločný formát znamená aktívne sa dohodnúť na takom formáte, ktorému príjemca rozumie
* Čo je možné vzájomne dojednať?
  + Video kódovanie
  + Hlasové kódovanie
  + Kódovanie diakritiky
  + ... atď.

Dojednanie spoločného formátu dát

* Ako príklad použijeme žiadosť WWW klienta o webovú stránku

GET / HTTP/1.1

Host: sme.sk

User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.13) Gecko/20080311 Iceweasel/2.0.0.13 (Debian-2.0.0.13-1)

Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5

Accept-Language: sk,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-2,utf-8;q=0.7,\*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Kódovanie diakritiky

* Pôvodná znaková sada pre počítače bola navrhovaná pre americkú abecedu a niektoré zvláštne znaky v textovom režime
  + ASCII – American Standard Code for Information Interchange
* Znaková sada však mohla obsahovať najviac 256 rôznych znakov
  + Národné znaky všetkých jazykov sa do tejto znakovej sady naraz nezmestia
* Preto vznikla séria rôznych znakových sád pre rôzne jazyky
* Znakové sady (niekedy nazývané aj kódové stránky):
  + ASCII
    - Americká abeceda
    - Definuje iba 127 znakov
    - Zobraziteľné znaky sú na pozíciách 33 až 126
  + ISO-8859-1
    - Americká abeceda s rozšírením o niektoré špeciálne znaky
    - Definuje 256 znakov
* Znakové sady (pokr.):
  + ISO 8859-2
    - Abeceda pre vybrané európske jazyky vrátane slovenčiny
    - Definuje 256 znakov, z toho 191 zobraziteľných
    - Niekedy sa táto kódová sada nazýva aj Latin 2
  + Windows 1250
    - Analogická kódová stránka ako ISO 8859-2
    - Navrhnutá nezávisle spoločnosťou Microsoft
    - 8859-2 a Windows 1250 nie sú kompatibilné
* Doposiaľ uvedené kódové stránky majú zásadný problém
  + Nedá sa jednoducho zariadiť, aby jeden dokument bol písaný v mnohých jazykoch, a teda v mnohých kódových stránkach
* Z tohto dôvodu vznikla iniciatíva vytvoriť takú kódovú stránku, ktorá bude obsahovať znaky všetkých jazykov
  + 256 možností nestačí
  + Odpoveďou na túto iniciatívu je štandard UNICODE,  
    v ktorom môže byť až 232 znakov

Ďalšie funkcie prezentačnej vrstvy

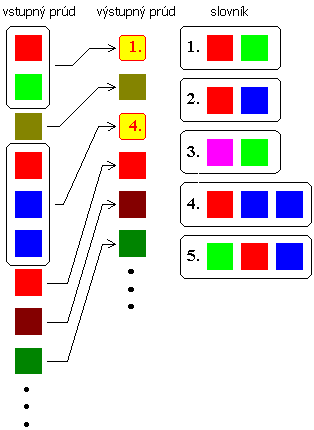
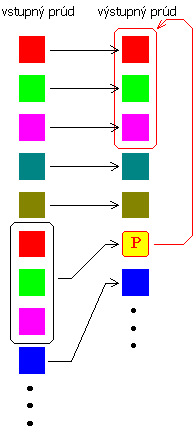
* Kompresia a dekompresia dát
  + Používateľ túto činnosť spravidla ani nevníma
  + Mnohokrát je súčasťou inej operácie, spravidla pri kódovaní hlasu alebo videa
  + Niektoré WWW servery dokážu v reálnom čase vyžiadanú WWW stránku skomprimovať a odoslať. Browser si ju transparentne dekomprimuje a zobrazí
* Šifrovanie a zabezpečenie prenášaných dát aplikáciou
  + Elektronická pošta s využitím digitálneho podpisu, prípadne  
    i zašifrovania
  + Zabezpečené WWW lokality – HTTPS
  + Prenos šifrovaného hlasu, videa (napr. Skype)

Kompresia dát

* Kompresia/komprimácia/pakovanie dát slúži na zmenšenie ich objemu
* Existujú principiálne dve základné kategórie kompresií
  + Bezstratová/nestratová: po kompresii a dekompresii získavame originálne dáta bez akejkoľvek zmeny
  + Stratová: po kompresii a dekompresii získavame pozmenené dáta – isté pôvodné časti sa stratia

LZW - autori Lempel Ziv Welch

* Využívajú to, že v komprimovaných dátach sa nachádzajú opakujúce sa sekvencie dát - Slovníkové metódy



Stratové kompresné metódy

* + Vychádzajú z predpokladu, že v originálnych dátach sa nachádzajú časti, vypustenie ktorých je možné tolerovať
    - Nepočuteľné detaily
    - Neviditeľné detaily
  + Bývajú optimalizované na konkrétny účel použitia a dosahujú vo všeobecnosti lepšie kompresné pomery ako bezstratové algoritmy

Kompresia videa

* Nekódujú sa celé snímky, ale len zmena medzi jednotlivými obrazmi
* Obraz sa delí na pixely a tie je možné deliť do špecifických skupín
* Štandardy
  + MPEG
  + H-261, H263

Audiou kompresia

* Vzorkovanie
* Vzorkovací kmitočet
  + 8.00 kHz Telefónia
  + 16 kHz Multimediálna komunikácia
  + 22.05 kHz Osobný počítač
  + 32.00 kHzDigitálne rádiové a televízne vysielanie
  + 44.10 kHzCD – Audio
  + 48.00 kHzDAT kazetové magnetofóny, HDTV96.00 kHzDVD
* ADPCM (Adaptive Differential Pulse Code Modulation), spočívajúc v odhadnutí vzorky podľa predchádzajúce vzorky, kvôli faktu, že u audio signálu je silná korelácia medzi nasledujúcimi vzorkami –bezstratová

Kompresia dát v client –server

* Originálne prenášané dáta (napr. súbor z pevného disku) nemusia byť pôvodne komprimované
* Komprimáciu a spätnú dekomprimáciou môžu v momente prenosu, teda v reálnom čase, zabezpečiť klient so serverom
* Žiadosť WWW klienta o webovú stránku

GET / HTTP/1.1

Host: sme.sk

User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.13) Gecko/20080311 Iceweasel/2.0.0.13 (Debian-2.0.0.13-1)

Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5

Accept-Language: sk,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-2,utf-8;q=0.7,\*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Komprimačné formáty

* Stratové
  + Audio: MP3,OGG, MP2, WMA, G.729, EFR ...
  + Video: MPEG, XVID, WMV, DIVX, Ogg Theora, Sorenson ...
  + Grafika: JPEG, GIF, ...
* Bezstratové
  + Dáta: zip, rar, arj, cab, tar, gz,

Realizácia komprimácie

* Komprimácia sa vykonáva
  + automaticky (uložením súboru v komprimovanom formáte JPG, MPEG, MP3)
  + pomocou špeciálneho komprimačného programu (ZIP, RAR) – archivačné formáty
* Poznámka
  + Komprimovať archivačným formátom sa oplatí len súbory, ktoré ešte neboli komprimované

Šifrovanie

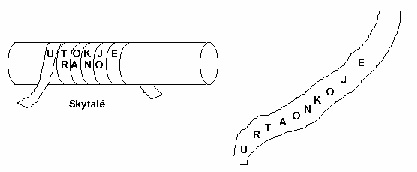
* je kryptografické pretváranie údajov, ktorého výsledkom je šifrovaný text
* Rozdiel medzi šifrovaním a kódovaním
  + Kódovanie pretvára údaje za účelom uľahčenia prenosu
  + Šifrovanie sa snaží o ich utajenie

Veda o šifrovaní

* Kryptografia - veda zaoberajúca sa prevedením informácie do nečitateľnej podoby
* Kryptoanalýza - veda zaoberajúca matematickými postupmi na získanie pôvodného obsahu zašifrovanej informácie

História šifrovania

* Staré Grécko palica skytalé



* Antický Rím – Cézarova šifra posun písmen- monoalfabetická šifra
* Polyalfabetická šifra - prvé písmeno sa šifruje poľa jednej substitúcie, druhé podľa inej substitúcie
* Systémy na matematickom základe – kryptografické systémy
  + DES - DATA ENCRYPTION STANDARD
  + RSA kryptosystém
  + Systémy/protokoly verejného kľúča

Model krypto systému

* Kryptografický systém sa skladá z:
  + šifrovacieho algoritmu
  + dešifrovacieho algoritmu
  + kľúča
* Algoritmy sa neutajujú – (iba z obchodného dôvodu), utajuje sa kľúč
  + Ak K1 = K2 SYMETRICKÁ KRYPTOGRAFIA
  + Ak K1 = K2 ASYMETRICKÁ KRYPROGARFIA

Základné typy krypto systémov

* Symetrické – jeden kľúč
  + Jedným kľúčom sa šifruje aj dešifruje
* Asymetrické – dvojica kľúčov
  + Každý účastník má dvojicu kľúčov
  + Čo sa jedným kľúčom zašifruje dá sa dešifrovať iba druhým
  + Jeden (súkromný) kľúč je bezpečne ukrytý
  + Druhý (verejný) kľúč je prístupný všetkým

Symetrické šifrovanie

Asymetrické šifrovanie

Asymetrické kryptosystémy

* pomalšie – zložité výpočty
* dohľad nad kľúčmi – certifikačná autorita
* nie je potrebné dohovárať si tajné kľúče, nemusia sa poznať

Digitálny/elektronický podpis

Systémy elektronického podpisu

* Sú na báze asymetrickej kryptografie
* Zaručujú autenticitu a integrity správy
* Vyjadrujú pravosť podpisu

Elektronický podpis

Overenie podpisu

Verejný kanál

Zneužitie

Načo certifikát ?

* Kvôli výmene verejných kľúčov
* CA vydáva certifikáty, ktoré obsahujú:
  + Údaje o vlastníkovi certifikátu
  + Verejný kľúč
  + Dobu platnosti certifikátu
  + Digitálny podpis CA
* CA musí mať certifikát podpísaný:
  + Sebou (Root CA)
  + Nadradenou CA

Použitie certifikátov

* Šifrovanie dokumentov, mail-ov ...
* Podpisovanie dokumentov, mail-ov ...
* Podpisovanie programov, driver-ov ...
* Informačné systémy (žiadny login/password)
* Zabezpečenie známych protokolov (https, pop3s, imaps, smtp,...)
* VPN, IPSEC
* ...