

# Java tečaj

8. dio

Distribuirane aplikacija.  
Paket java.net.

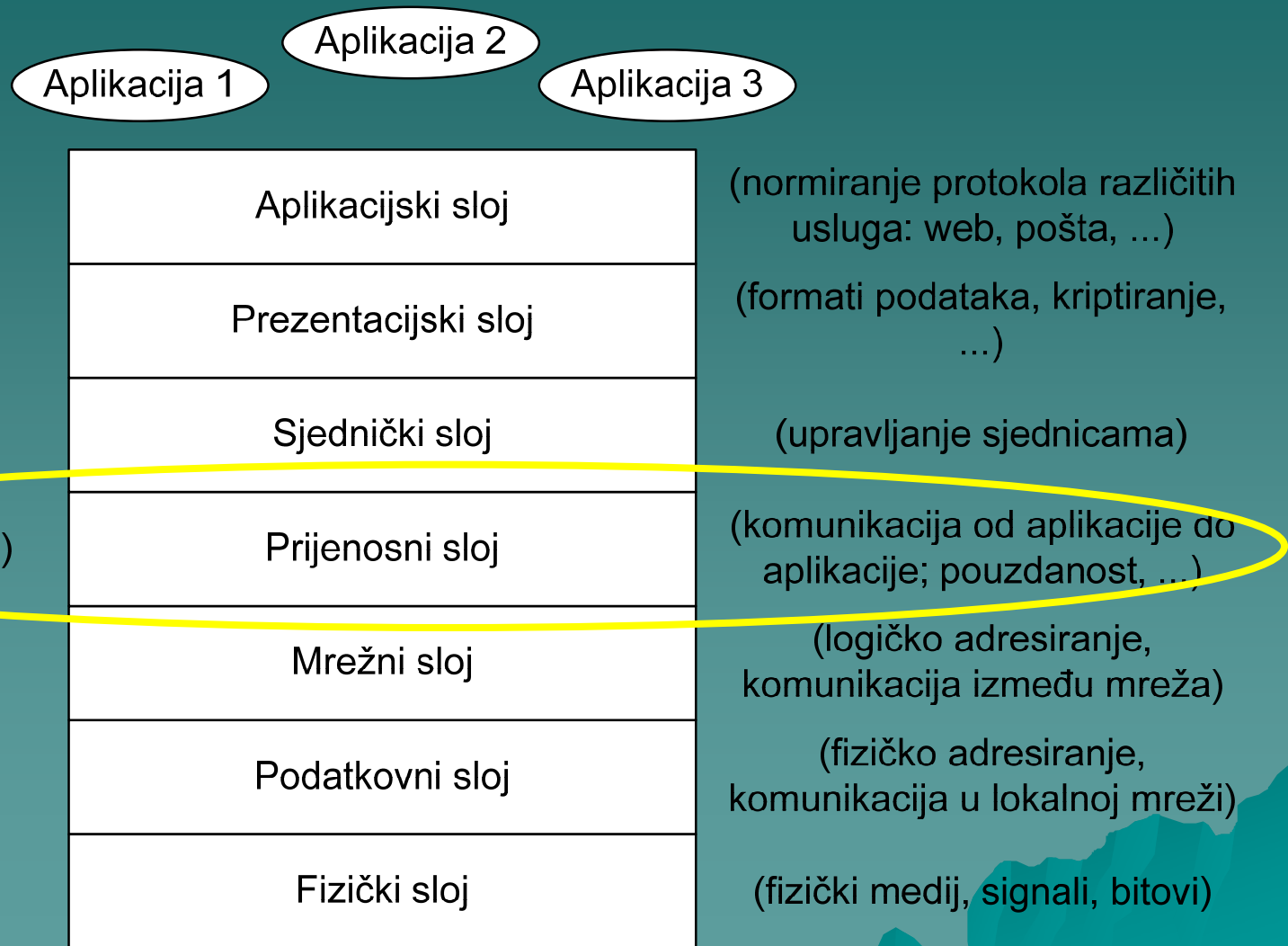
© 2012.

A stylized, dark teal silhouette of a mountain range is located in the bottom right corner of the slide, partially overlapping the copyright text.

# Uvod

- ◆ U današnje doba računala su umrežena
  - Korištenje mrežnih resursa (printeri, ...)
  - Web (HTTP, HTTPS)
  - Elektronička pošta (POP3, IMAP, SMTP)
  - Razmjena datoteka (FTP, P2P, ...)
  - Rad na udaljenom računalu (ssh, ...)
- ◆ Internet

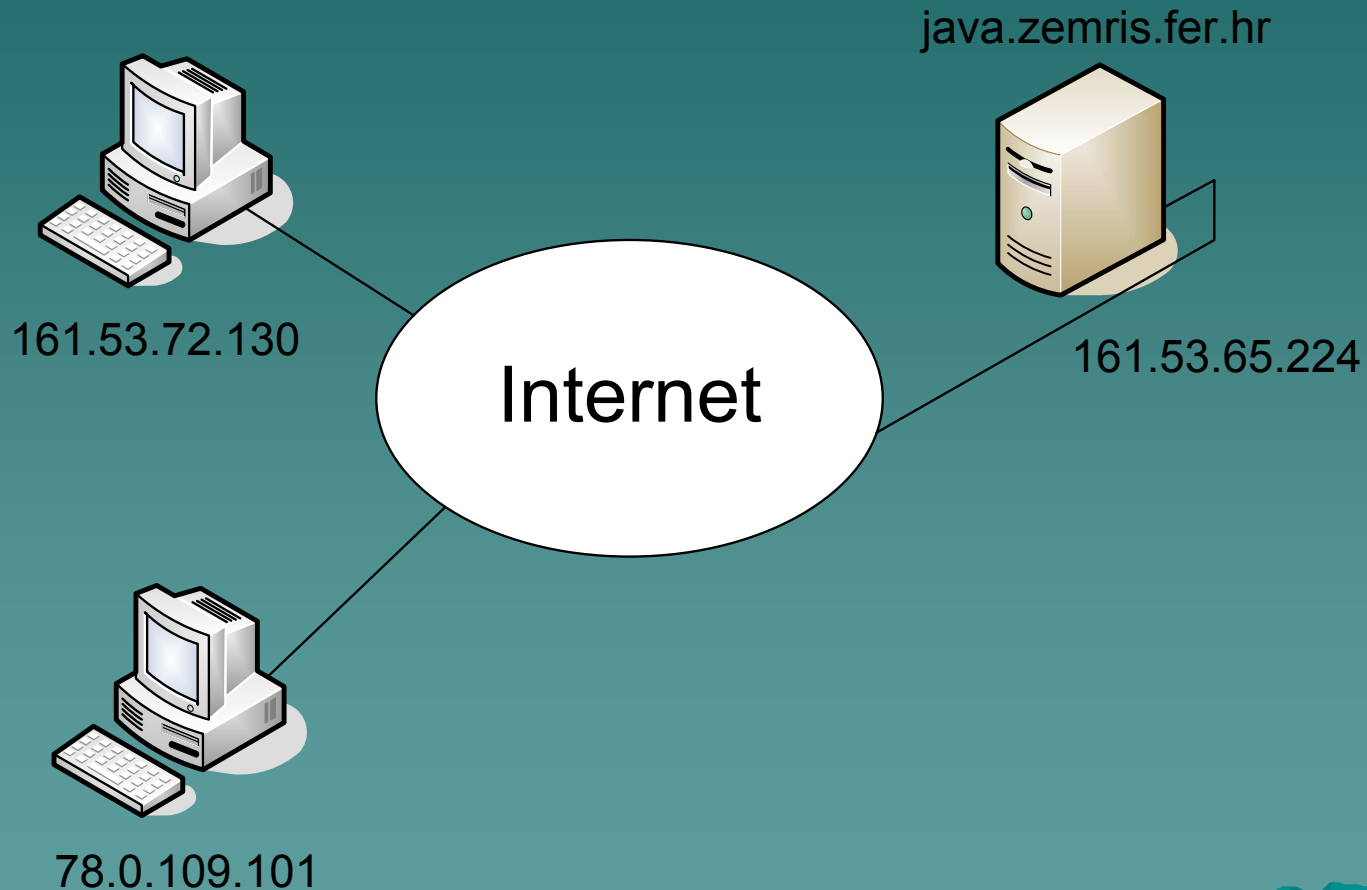
# Uvod: ISO/OSI referentni model



# Protokoli mrežnog sloja

- ◆ Nužnost: računala međusobno moraju moći komunicirati!
- ◆ Danas dominantno: IP protokol
- ◆ Svako računalo ima svoju **jedinstvenu** IP adresu
- ◆ IP protokol prenosi pakete "viših" protokola od jednog računala do drugog (zahvaljujući IP adresama)

# Protokoli mrežnog sloja



# Protokoli mrežnog sloja

- ◆ Koja je IP adresa Vašeg računala?
- ◆ Na Windowsima
  - Otvorite Command Prompt
  - Zadajte naredbu ipconfig
  - Potražite Local area connection (ili kako se već spajate na Internet)

# Protokoli mrežnog sloja

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\marcupic>ipconfig

Windows IP Configuration

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.40.1
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.227.1
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 

Ethernet adapter Local Area Connection:

    Media State . . . . .             : Media disconnected

Ethernet adapter Wireless Network Connection:

    Connection-specific DNS Suffix  . : localdomain
    IP Address. . . . .               : 192.168.1.7
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 192.168.1.10

C:\Documents and Settings\marcupic>
```

# Protokoli mrežnog sloja

- ◆ Koja je IP adresa Vašeg računala?
- ◆ Na Linuxu
  - Otvorite terminal
  - Zadajte naredbu `ip addr sh`
  - Potražite `eth0`, `ppp0` (ili kako se već spajate na Internet)



# Protokoli mrežnog sloja

```
marcupic@sun:~  
[marcupic@sun ~]$ /sbin/ip addr sh  
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 00:0b:6a:e3:fc:70 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.10/24 brd 192.168.1.255 scope global eth0  
    inet6 fe80::20b:6aff:fee3:fc70/64 scope link  
        valid_lft forever preferred_lft forever  
3: sit0: <NOARP> mtu 1480 qdisc noop  
    link/sit 0.0.0.0 brd 0.0.0.0  
4: vmnet1: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 00:50:56:c0:00:01 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.212.1/24 brd 192.168.212.255 scope global vmnet1  
    inet6 fe80::250:56ff:fec0:1/64 scope link  
        valid_lft forever preferred_lft forever  
26: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP> mtu 1492 qdisc pfifo_fast qlen 3  
    link/ppp  
    inet 78.0.109.29 peer 172.29.252.47/32 scope global ppp0  
[marcupic@sun ~]$
```


# Protokoli mrežnog sloja

- ◆ IP protokol možemo gledati kao “tačke Interneta”
  - Prijenos podataka bilo kojeg računala na Internetu do bilo kojeg računala na Internetu; neovisno o zemljopisnom položaju, državi, gradu i sl.
  - Ali ništa više! IP protokol ne razlikuje aplikacije!

# Protokoli mrežnog sloja

- ◆ IP od računala do računala prenosi podatke – “teret”
- ◆ IP ne tumači što je taj teret
  - Tipično, to su podaci protokola viših slojeva o kojima IP ne zna ništa!

# Protokoli prijenosnog sloja

- ◆ Kako bi se omogućila komunikacija aplikacija – aplikacija, koriste se protokoli prijenosnog sloja
  - ◆ Danas tipično TCP ili UDP
  - ◆ Ovi protokoli u pakete dodaju dvije važne informacije: izvorišni port i odredišni port
  - ◆ Za prijenos poruka koriste IP
- 
- A stylized, dark teal mountain range graphic is located in the bottom right corner of the slide, partially overlapping the text area.

# Protokoli prijenosnog sloja

- ◆ Prijenosni sloj: UDP, TCP

- ◆ UDP

- Paketno orijentirani protokol:

- ◆ Direktno radimo s datagramima

- Nema nikakvih garancija

- ◆ Isporuka?

- ◆ Redoslijed?

- ◆ Dupliciranje datagrama?

# Protokoli prijenosnog sloja

- ◆ Prijenosni sloj: UDP, TCP
- ◆ TCP
  - Tokovno orijentirani protokol:
    - ◆ Radimo s tokovima okteta
    - ◆ Protokol se sam brine za prijenos
  - Nude se garancije
    - ◆ Isporuka?
    - ◆ Redoslijed?
    - ◆ Dupliciranje?

# Model klijent-poslužitelj

- ◆ Najjednostavniji model mrežne aplikacije jest model klijent – poslužitelj
  - Poslužitelj je aplikacija koja nudi određene usluge
  - Čeka na unaprijed poznatom portu zahtjeve, i kada stignu, šalje odgovore

# Model klijent-poslužitelj

- ◆ Najjednostavniji model mrežne aplikacije jest model klijent – poslužitelj
  - Klijent je aplikacija koja traži određene usluge
  - Spaja se na poslužitelj na unaprijed definirani port, šalje zahtjeve, i čeka na odgovore



# Model klijent-poslužitelj

- ◆ Pri tome, ovisno o korištenom protokolu, klijentu i poslužitelju se može garantirati sigurna isporuka poruka (TCP), ali i ne mora (UDP; što tada?)

# Model klijent-poslužitelj

- ◆ Jedan od najpoznatijih primjera primjene UDP-a i modela klijent – poslužitelj: DNS protokol
  - Razrješava simboličko ime računala u IP adresu
- ◆ Probajte u konzoli upisati:

```
nslookup java.zemris.fer.hr
```

# Model klijent-poslužitelj

## ◆ Važno zapamtiti!

- Na jednom računalu može biti više poslužitelja (aplikacija), ali svaki poslužitelj čeka na svojem portu!
- Portovi manji od 1024 mogu se koristiti samo uz posebne dozvole (primjerice, to može root ili Administrator)
- Važno: danas je na računalu uobičajeno prisutan i *firewall* koji onemogućava rad poslužitelja pa njegov razvoj zna biti problematičan

# Model klijent-poslužitelj

- ◆ Konkretna implementacija komunikacijskih protokola nudi sam operacijski sustav
- ◆ Aplikacije od OS-a traže “pristupnu točku” (socket) kroz koji šalju/primaju podatke

# UDP protokol

- ◆ UDP protokol između klijenta i poslužitelja šalje pakete
  - Datagram je najmanja jedinica podataka koja se šalje
  - Da bi poslali UDP datagram, trebamo zadati IP adresu i port odredišta (kome šaljemo?) te IP adresu i port pošiljatelja (tko šalje?)

# UDP protokol

- ◆ Napravimo poslužitelj koji prima UTF-8 kodiran tekst oblika:  
"x + y" ili "x - y"  
te koji klijentu vraća izračunati rezultat isto tako kao tekst ili u slučaju pogreške vraća tekst "ERR"
- ◆ Napravimo klijenta koji korisnika pita za izraz, šalje ga poslužitelju i ispisuje rezultat

# UDP protokol

- ◆ U Javi, sve potrebno nalazi se u paketu `java.net`
- ◆ Datagram koji se šalje modeliran je razredom `DatagramPacket`:

```
DatagramPacket(  
    byte[] buf,  
    int len  
)
```

# UDP protokol

- ◆ Datagramu je potrebno zadati IP adresu odredišnog računala te port (`setAddress(...)`, `setPort(...)`)
- ◆ Za dobivanje primjerka razreda `InetAddress` može se koristiti poziv:

```
InetAddress.getByName(  
    "imeIliIP"  
)
```



# UDP protokol

- ◆ Pristupnu točku za ovaj protokol modelira razred `DatagramSocket`
- ◆ Napišimo poslužitelja i klijenta i isprobajmo ih!
- ◆ Datagram se šalje metodom `send`, a prima metodom `receive`
- ◆ Metoda `receive` očekuje referencu na datagram u koji će iskopirati pristiglo

# TCP protokol

- ◆ Koriste se:
  - `ServerSocket` za poslužiteljsku stranu
  - `Socket` za klijentsku stranu
- ◆ `Socket` ima metode za razmjenu poruka (uočite – ne datagrama!)
  - `getInputStream()`
  - `getOutputStream()`

# TCP protokol

- ◆ Kao primjer napišimo klijenta i poslužitelja za isti problem kao i u prethodnom zadatku!
- ◆ Neka se šalje upit po upit kao pojedinačni retci (s \n na kraju)
- ◆ Posebna naredba "quit" poslana od klijenta znači da treba prekinuti uspostavljenu vezu

# TCP protokol

- ◆ Tipičan primjer ilustracije protokola TCP je HTTP poslužitelj
- ◆ Na sljedećim slideovima su prikazani upit i odgovor protokolom HTTP

# HTTP zahtjev

Npr. za URL: `http://localhost:6677/dir1/index.html`

```
GET /dir1/index.html HTTP/1.1
Host: localhost:6677
User-Agent: Mozilla/5.0 ... Firefox/2.0.0.3
Accept: */*
Accept-Language: en-us,hr;q=0.7,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

# HTTP odgovor

Npr. za URL: `http://localhost:6677/dir1/index.html`

HTTP/1.1 200 OK

Content-Length: 76831

Content-Type: text/html

<html>

...

</html>