

# World Wide Web: Lato server

M. Colajanni, R. Lancellotti, F.Mazzoni

## Componenti di un sito Web

- Piattaforma Hardware
- Software di base
- Server Web (collega piattaforma alla parte informativa)
- Parte informativa del sito Web

M. Colajanni, R. Lancellotti, F.Mazzoni

## Parte informativa

- Risorse ipermediali con link verso risorse esterne
- Organizzazione ad albero
  - Assomiglia ad un filesystem gerarchico
- Ogni pagina ha un nome unico corrispondente al path assoluto rispetto alla radice dell'albero

M. Colajanni, R. Lancellotti, F.Mazzoni

## Organizzazione delle pagine

- Non è necessario che l'organizzazione delle pagine rispecchi quella fisica nel filesystem
  - Es. Alias
  - Distribuzione su più partizioni, anche condivise in rete...

M. Colajanni, R. Lancellotti, F.Mazzoni

## Tipi di pagine

- In base al contenuto
  - Pagine HTML
  - Immagini
  - ...
- Classificazione funzionale
  - Pagine statiche
  - Pagine volatili
  - Pagine dinamiche
  - Pagine attive

M. Colajanni, R. Lancellotti, F. Mazzoni

## Il server Web

- Client e server comunicano mediante TCP
- Gestione TCP a cura del S.O.
- Come interagiscono Web server e S.O.?
- API Application Program(ming) Interface (System Call)

M. Colajanni, R. Lancellotti, F. Mazzoni

# API

- Parte del Sistema Operativo
- Definiscono:
  - Operazioni consentite
  - Argomenti delle operazioni
- Socket API
  - Definite in BSD Unix
  - Standard

M. Colajanni, R. Lancellotti, F.Mazzoni

# Socket

- Meccanismo IPC (Inter-Process Communication) remoto
- Strumento base per servizi di rete
- Astrazione dei processi sottostanti (nascondono dettagli di basso livello di TCP e UDP)

M. Colajanni, R. Lancellotti, F.Mazzoni

## Fasi della comunicazione

- Si informa il S.O. che si vuole instaurare una connessione
- Apertura della connessione
  - Lato server: si attende una connessione e la si accetta
  - Lato client: si suppone che il server sia in attesa
- Scambio di dati
- Chiusura della connessione

M. Colajanni, R. Lancellotti, F.Mazzoni

## Inizializzazione

- `int socket (int domain, int type, int protocol);`
  - Prima istruzione eseguita
  - Inizializza strutture dati per gestire la connessione
  - Ritorna file descriptor o -1
  - Il client usa il socket direttamente
  - Il server lo usa come prototipo per i socket che effettivamente saranno usati

M. Colajanni, R. Lancellotti, F.Mazzoni

## Parametri di socket

- Domain (famiglie di protocolli)
  - AF\_INET, AF\_UNIX
- Type
  - SOCK\_STREAM, SOCK\_DGRAM, SOCK\_RAW
- Protocol
  - Tipicamente lasciato a 0, seleziona un protocollo qualora ci siano ambiguità

M. Colajanni, R. Lancellotti, F.Mazzoni

## Apertura connessione

- Chiamata a lato client
- `int connect(int sock_fd, struct sockaddr *serv_addr, int addrlen);`
- Si usa con SOCK\_STREAM
- Server in attesa
- `*serv_addr` contiene IP address e numero di porta del server
- Ritorna -1 in caso di errore

M. Colajanni, R. Lancellotti, F.Mazzoni

## Trasmissione dati

- `int send(int sock_fd, const void *buf, int len, unsigned int flags);`
- Dati da spedire in `*buf`
- Si usa solo con `SOCK_STREAM`
- Ritorna numero bytes inviati o -1
- Flags: tipicamente impostati a 0
- Alternative: `sendto()`, `sendmsg()`, usabili anche per `SOCK_DGRAM`

M. Colajanni, R. Lancellotti, F.Mazzoni

## Ricezione dati

- `int recv(int sock_fd, void *buf, int len, unsigned int flags);`
- Solo per socket connessi
- Legge un messaggio di lunghezza massima `len`
- Ritorna num. byte letti o -1
- Flags tipicamente lasciato a 0

M. Colajanni, R. Lancellotti, F.Mazzoni

## Chiusura

- `int close(int sock_fd);`
- Usata sia nel server che nel client
- Chiude una connessione (SOCK\_STREAM)
- La connessione è effettivamente chiusa quando sia il client che il server chiudono

M. Colajanni, R. Lancellotti, F.Mazzoni

## Assegnare un nome

- `int bind(int sock_fd, struct sockaddr *my_addr, int addrlen);`
- Collega un socket ad un indirizzo visibile dall'esterno.
- Si usa tipicamente dal lato server

M. Colajanni, R. Lancellotti, F.Mazzoni



## Specifica del backlog

- `int listen(int sock_fd, int backlog);`
- Specifica al S.O. la dimensione di una coda di richieste in attesa su una socket
- Se il server sta servendo una richiesta e ne arriva un'altra questa viene accodata
- Se la coda è piena la connessione viene rifiutata
- Solo per connessioni!

M. Colajanni, R. Lancellotti, F.Mazzoni

## Accept

- `int accept(int sock_fd, struct sockaddr *addr, int addrlen);`
- Tipicamente bloccante
- `*addr` IP address e numero porta del chiamante
- Ritorna una nuova socket che verrà usata per il servizio della connessione

M. Colajanni, R. Lancellotti, F.Mazzoni

## Puntatori per approfondimenti

- D. E. Comer "Computer Networks and Internet"
- W. R. Stevens "Unix Network Programming" Vol. 1
- Man pages...

M. Colajanni, R. Lancellotti, F. Mazzoni