Redes 2

Camada de Aplicação - Continuação

- Socket
    - Abrindo um Socket
        - Aloca uma entrada na tabela de descritores e cria a estrutura de dados correspondente
        - sock_desc = socket(AF_INET, transport, protocol)
        - AF_INET: Address Family InterNET
        - transport: SOCK_STREAM para TCP
        - transport: SOCK_DGRAM para UDP
        - protocol: 0 para familia de protocolos TCP/IP
    - Fechando um Socket
        - Libera a estrutura de dados e a entrada na tabela de descritores
        - close(sock_descr)
    - Servidor Especifica Porta Local
        - Servidor especifica a porta conhecida do serviço: bind(sock_descr, enderec_local, sizeof(enderec_local))
        - Para especificar endereços: vamos usar um registro da biblioteca socket: sockaddr_in
        - Tem campos para endereço IP, porta, protocolo, etc.
    - Cliente Especifica Endereço Remoto
        - Cliente especifica endereço do servidor: connect(sock_descr, enderec_serv, sizeof(enderec_serv))
        - Para especificar enderec_serv: sockaddr_in
        - UDP pode usar connect: apenas armazena, específica endereço remoto dos datagramas que vai enviar
        - No caso do TCP: estabelece conexão
    - Transmissão de Dados
        - A função default para ser usada write(sock_descr, buffer, num_bytes_transmitir)
        - 4 alternativas
            - send: permite que o programador manipule flags que controlam a transmissão, por exemplo: dados urgentes
            - writev: especifica um vetor de apontador para os dados a serem transmitidos
            - sendto: UDP, permite especificar o endereço do destinatário
            - sendmsg: todas as possibilidades acima
    - Recebendo Dados
        - A função default para ser usada read(sock_descr, buffer, num_bytes_máximo)
        - 4 alternativas
            - recv: permite que o programador manipule flags que controlam a transmissão, por exemplo: dados urgentes
            - readv: especifica um vetor de apontador para os dados a serem transmitidos
            - recvfrom: UDP, permite especificar o endereço do destinatário
            - recvmsg: todas as possibilidades acima

- Servidor: Escuta e Conexão
    - Um servidor fica em modo de escuta: só executa uma vez
      listen(sock_descr, tam_fila)
    - Tamanho sugerido para a fila de clientes: 5
    - Parece pouco, mas
        - Nos servidores concorrentes: cliente só fica até abrir o processo filho para atendê-lo
        - Nos servidores iterativos: deve ser suficiente, caso contrário transforme em concorrente
    - Para estabelecer uma conexão solicitada pelo cliente
      accept(sock_descr, enderec_cli, sizeof(enderec_cli))
    - Para especificar enderec_cli
      sockaddr_in