Redes 2

Camada de Aplicação - Continuação

- Cliente TCP
  - ```
    Cliente Orientado à Conexão

    1. Determine o endereço IP e porta do
    servidor com o qual deseja se comunicar
    2. Abra um socket usando uma porta local
    livre para comunicar
    3. Estabeleça a conexão com o servidor
    4. Comunique-se, de acordo com o
    protocolo de aplicação
    5. Encerre a conexão e o socket
    ```
- Servidor TCP Iterativo
  - ```
    Servidor TCP Iterativo

    1. Abra um socket usando a porta conhecida
    para o serviço
    2. Fique à escuta, aguardando requisições
    de conexão de clientes
    3. Aceite requisição de conexão com o
    cliente: abra um novo socket para atendê-lo
    4. Comunique-se, de acordo com o protocolo
    de aplicação
    5. Encerre a conexão e volte ao passo 3
    ```
- Servidor UDP Iterativo
  - ```
    Servidor UDP Iterativo

    1. Abra um socket usando a porta conhecida
    para o serviço
    2. REPITA: receba requisição do cliente,
    formule resposta e envie de volta ao
    cliente, de acordo com o protocolo de
    aplicação
    ```
- Cliente UDP
  - ```
    Cliente Não Orientado à Conexão

    1. Determine o endereço IP e porta do
    servidor com o qual deseja se comunicar
    2. Abra um socket usando uma porta local
    livre para comunicar
    3. Comunique-se, de acordo com o
    protocolo de aplicação
    5. Encerre o socket
    ```
- Servidor TCP Concorrente
  - ```
    Servidor TCP Concorrente
    PAI
    1. Abra um socket usando a porta conhecida
    para o serviço
    2. Fique à escuta, aguardando requisições de
    conexão de clientes
    3. Aceite requisição de conexão com o cliente:
    abra um processo filho para atendê-lo
    FILHO
    1. Estabeleça a conexão com o cliente, atenda-
    o em um novo socket
    2. Comunique-se, de acordo com o protocolo de
    aplicação
    3. Encerre a conexão e o processo
    ```
    ```

- Servidor Concorrente: Múltiplos Processos
    - Todos os processos se comunicam na mesma porta conhecida para o serviço
    - Assim, as portas são usadas na identificação dos processos que se comunicam
    - As portas são usadas na identificação, mas não são os identificadores
    - No caso de servidores concorrentes: o S.O. usa o (endereço IP, porta) do cliente para identificar para qual processo é cada mensagem
    - Se está conectado: mensagem é para o processo filho correspondente
    - Se não está conectado: mensagem é para o pai
- Sockets
    - Não é um padrão TCP: não há RFC para sockets
    - Não há exclusividade
    - É possível usar sockets para a comunicação usando outra pilha de protocolos
    - É possível fazer programação de sistemas TCP/IP usando outras API's
    - Como API
        - Interface para programar algo, no caso comunicação de dados
        - Conjunto de funções: você inclui uma biblioteca e tem acesso a todas as funções necessárias para construir programas que se comunicam usando TCP/IP
    - No Linux
        - Entidades da mesma natureza que processos, arquivos, etc.
        - O identificador do socket é chamado de descritor
        - O descritor é um inteiro que indexa uma tabela de apontadores para estruturas de dados com informações sobre o socket