

A Camada de Aplicações

Adriano, Amanda, Douglas, Guilherme e Larissa.

Sumário

A Camada de aplicações

9.1 Temas de projeto da camada de aplicações.

9.2 Transferência, acesso e gerenciamento de Arquivos.

9.3 Correio Eletrônico.

9.4 Terminais Virtuais.

9.5 Outras aplicações.

9.6 Exemplo da camada de aplicações.

9.1 A Camada de Aplicações

Descrição

9.1.1 Transferência, Acesso e Gerenciamento de Arquivos

9.1.2 - Correio eletrônico

9.1.3 - Terminais Virtuais

9.1.4 - Outras aplicações

9.1.5 - Elementos de Serviços de OSI - ACSE e CCR

Descrição

- Última camada do modelo OSI.
- Responsável por oferecer serviços de comunicação para os **programas de usuário**.

9.1.1 Transferência, Acesso e Gerenciamento de Arquivos

- Transferência e acesso a arquivos remotos
- Pode operar de duas maneiras:
 - Manter o arquivo original em um equipamento e transferir suas cópias para outras máquinas;
 - Manter arquivo ativo na sua máquina de criação e enviar cópias quando necessário.
- Armazenamento de arquivos virtuais
 - programas de aplicação podem acessar e transferir arquivos remotos sem total conhecimento dos servidores.

9.1.2 Correio Eletrônico

- ARPANET
- Características que o diferenciam:
 - rápido
 - uma mensa mensagem pode ser enviada a várias pessoas ao mesmo tempo;
 - receptores finais normalmente são humanos;
 - não exige que as duas partes estejam disponíveis;
 - interface para transporte posta;
 - documentos altamente estruturados.

9.1.3 Terminais Virtuais

- Sequência de Escape
 - sequência de caracteres para manipular o terminal
- Estrutura de dados abstrata
 - pode ser manipulada pelo teclado e computador, e tem estado atual refletido no monitor
- Representa o estado obtido do terminal real

9.1.4 Outras Aplicações

- Serviço de Diretório
 - fornece um meio para encontrar o endereço de rede de pessoas e serviços disponíveis na rede
- Entrada Remota de Jobs
 - permite a um usuário trabalhando em um computador controlar a execução de um job em outro computador.
- Telemática
 - serviços de informação pública para uso no lar e no escritório.

9.1.5 Elementos de Serviços de OSI - ACSE e CCR

- Padronização da gerência de conexões e da coordenação de atividades entre três ou mais partes.
- Usa blocos básicos de construção.

Elemento de Serviços de Controle de Associação

- ASCE (Association Control Service Element)
- Gerencia conexões
- Mapeamento de um para um entre as primitivas.

Primitiva ACSE do OSI	Descrição
A-ASSOCIAR	Estabelece uma associação
A-LIBERAÇÃO	Libera uma associação
A-ABORTA	Encerramento iniciado pelo usuário
A-F-ABORTA	Encerramento iniciado pelo fornecedor

Figura 9-1 As primitivas do ACSE

Compromisso, Concorrência e Recuperação

- CCR (Commitment, Concurrency and Recovery).
- Imune a falhas.
- Realização de ações por completo
 - sucesso completo ou não faz absolutamente nada.
- Duas fases
 - Verificação se tarefa pode ser cumprida por todos os escravos.
 - Cumprir ou abortar ação.

MESTRE

```
Iniciar ação atômica;  
Enviar Pedido 1;  
. . .  
Enviar Pedido n;  
Enviar mensagem "Preparar para cumprir";  
  
if todos os escravos disseram "OK"  
    then Enviar mensagem "Cumprir";  
else  
    Enviar mensagem "Retornar";  
Esperar pelas confirmações
```

ESCRAVO

```
if ação pode ser realizada  
    then begin  
        Bloquear dados;  
        Armazenar estado inicial em disco;  
        Armazenar pedidos em disco;  
        Enviar mensagem "OK";  
    end  
  
else  
    Enviar mensagem "Falha";  
  
if mestre disse cumprir  
    then begin  
        Fazer trabalho;  
        Desbloquear dados;  
    end  
Enviar mensagem "Confirmação";
```

Figura 9-2 O comprometimento de duas fases

Primitiva CCR do OSI	De parte de	Descrição
C-INICIAR	MESTRE	Inicia uma ação atômica
C-PREPARAR	MESTRE	Fim da fase 1; prepara para registrar
C-PRONTO	ESCRAVO	O escravo está apto a fazer seu trabalho
C-RECUSAR	ESCRAVO	O escravo não é capaz de fazer seu trabalho
C-REGISTRAR	MESTRE	Cumpre a ação
C-RETORNAR	MESTRE	Aborta a ação
<i>C-REINICIAR</i>	<i>Qualquer um</i>	<i>Anuncia que ocorreu uma falha</i>

Figura 9-3 As primitivas do CCR

Transferência, Acesso e Gerenciamento de Arquivos - 9.2

Temas Abordados

- Projeto de Servidores de Arquivos - 9.2.1.
- O Controle da Concorrência - 9.2.2.
- Duplicação de Arquivos - 9.2.3.
- Algumas Questões de básicas de Implementações - 9.2.4.

Projeto de Servidores de Arquivos - 9.2.1.

As principais propriedades de um servidor de arquivos

1. Estrutura de arquivos.
2. Atributo de arquivo.
3. Operações de arquivo.

Projeto de Servidores de Arquivos - Estrutura de Arquivo - 9.2.1

- **Não estruturado:** é o arquivo inteiro ou massa não estruturada
- o servidor não tem informações sobre a estrutura.

Funções leitura e escrita do arquivo inteiro

- **Arquivo simples** sequência ordenada do registro, tamanho pode variar, registros podem ter chaves associados a eles. É possível que ter endereçamento por chave ou por posição, ou seja, pode ter suporte a registros individuais.

Funções: estendê-los, alterações ou deletar.

Projeto de Servidores de Arquivos - Estrutura de Arquivo -

9.2.1.

Arquivo hierárquico(ou árvore): cada nó pode ter uma chave ou registro de dados

É bom que os nós estejam ordenados por que ajuda em nó não rotulados.

Funções: leitura de arquivo inteiro se estiver ordenado em arquivo não rotulado, por posição tem alteração, delete.

Projeto de Servidores de arquivos(9.2.1).

ilustra um arquivo não-estruturado.

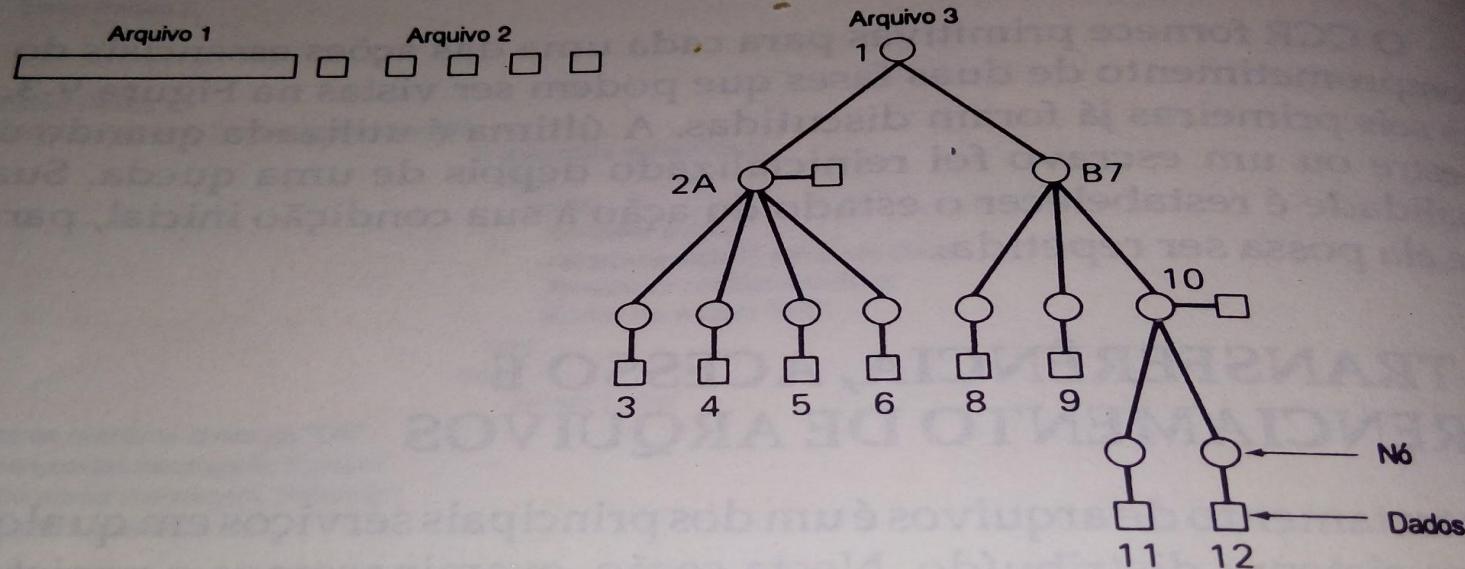


Figura 9-4 Arquivos (a) Não-estruturado (b) Seqüência de registros (c) Hierárquico

Projeto de Servidores de Arquivos - Atributo - 9.2.1.

Todos arquivos tem os atributos que os descrevem. e no mínimo, cada atributo deve ter um nome ou outro identificador, e um tamanho, para informar a quantidade de memória armazenada que ocupa.

Atributo	Tipo	Ativo quando arquivo é criado	Modificável pelo sistema	Mantido pelo sistema
Nome de Arquivo	String	X	X	
Operações permitidas	Mapa de bits	X		
Controle de acesso	Lista		X	
Número de conta	Inteiro	X	X	
Data e hora da criação de arquivo	Tempo	X		
Data e hora da última modificação em arquivo	Tempo	X		X
Data e hora da última leitura em arquivo	Tempo	X		X
Data e hora da última modificação de atributo	Tempo	X		X
Proprietário	Id de usuário	X		
Identidade do último modificador	Id de usuário	X		X
Identidade do último leitor	Id de usuário	X		X
Identidade do último modificador de atributo	Id de usuário	X		X
Disponibilidade de arquivo	Booleano	X	X	
Tipo de conteúdo	Id de objeto	X	X	
Chave de encriptação	String	X		
Tamanho	Inteiro	X		X
Tamanho futuro máximo	Inteiro	X	X	
Qualificações válidas	String	X	X	
Uso privativo	String	X	X	

Figura 9-5 Os atributos do armazenamento de arquivos virtual do OSI

Projeto de Servidores de Arquivos - Atributo - 9.2.1.

- Atributo: no mínimo tem um nome ou outro identificador.
- exemplo figura 9-5: mostrando os atributos do armazenamento de arquivos virtual do OSI, cada atributo tem um nome, tipo e um valor.
- o valor depende do servidor, da alteração do usuário e da criação do arquivo.
- O servidor tem acesso a data de arquivo, quem acessou e tamanho
- usuário o tem acesso ao controle do arquivo
- na criação do arquivo sao passadas as informações do arquivo.
- obs: mapa de bits, tem haver com proteção do arquivo.

Projeto de Servidores de Arquivos - Operações - 9.2.1.

Operações de arquivo - controle de acesso e proteção ao arquivo

- As Operações em arquivos podem ser aplicadas ao arquivo como um todo ou
- aos registros individuais, dependendo da estrutura de arquivo utilizada pelo servidor.
- além de operações em arquivo alguns servidores também suportam operações em diretórios, em que é possível criar e eliminar diretórios e mover arquivos de um diretório para outro.

Projeto de Servidores de Arquivos - Operações - 9.2.1.

- Simples - considerar todas máquinas seguras e dignas de confiança, e simplesmente executa todos os pedidos que chegarem.
- em computadores pessoais LAN é complicado.
- um solução: é incluir uma senha a cada transmissão, pelo uso de assinatura digital, usando a camada de apresentação.

Projeto de Servidores de Arquivos - Operações - 9.2.1.

Em serviço com conexão funciona: passando o endereço do chamador,

Operação	Aplica-se a arquivo	Aplica-se ao conteúdo	Mapa de bits	Descrição
Criar arquivo	X			Cria um novo arquivo
Eliminar arquivo	X	X		Destrói um arquivo existente
Selecionar arquivo	X		X	
Encerrar seleção de arquivo	X	X		Extrai um arquivo para controle de seus atributos
Abrir arquivo	X			Termina a seleção atual
Fechar arquivo	X	X		
Ler atributo	X		X	Abre um arquivo para leitura ou modificação
Alterar atributo	X	X	X	Fecha um arquivo aberto
Localizar		X		Lê um atributo de arquivo
Ler		X	X	Modifica um atributo de arquivo
Inserir		X	X	Localiza um registro
Substituir		X	X	Lê dados do arquivo
Estender		X	X	Insere novos dados no arquivo
Apagar		X	X	Sobrescreve dados existentes
				Acrescenta dados a algum registro
				Elimina um registro

Figura 9-6 As operações do armazenamento de arquivos virtual do OSI

Controle de concorrência - 9.2.2

- Um Servidor com vários clientes que acessam arquivos em comum, quando tem 2 ou mais clientes já tem um processo de concorrência.
- um exemplo é quando um cliente A tentar abrir um arquivo x para escrita mas aparece um cliente b que também faz escrita no arquivo x. o Arquivo B irá sobrepor as informações de A.

Controle de concorrência - 9.2.2

- **bloqueio compartilhado** se o arquivo estiver desbloqueado(livre) ou um bloqueio compartilhado sobre ele.
- o arquivo será usado para leitura para 1 ou mais pessoas(não tem, problema de concorrência em leitura), é garantido o pedido de abertura de bloqueio compartilhado e não vai ter abertura de escrita.
- **bloqueio exclusivo** somente quando arquivo estiver desbloqueado, somente uma pessoa acessa o arquivo e faz o pedido de escrita no arquivo.

Controle de concorrência - 9.2.2

Rejeição imediata cancelar todos pedidos, e fazer um bloqueio exclusivo.

é usado quando tem um intenso uso de leitura por muitos clientes e alguém quer fazer um bloqueio exclusivo(escrita) no arquivo mas praticamente o pedido dele nunca será aceito.

enfileiramento em fila enfileirar pedidos de leitura e escrita, depois dos leitores atuais terminarem o pedido do primeiro escritor da fila será aceito, podendo ou não ter outro escritor logo em seguida.

observações o enfileiramento em fila é uma política do sistema de arquivo, ou seja depende da sua implementação de projeto.

Controle de concorrência - 9.2.2

Problemas de Bloqueio

bloquear arquivos inteiro é possível não só apenas bloquear arquivos inteiros mas registros ou subárvores específicas.

clientes que tiver pane em seu bloqueios uma solução é interromper todos seus bloqueios.

Modelo são a leitura ou a escrita
ilustra um arquivo não-estruturado.

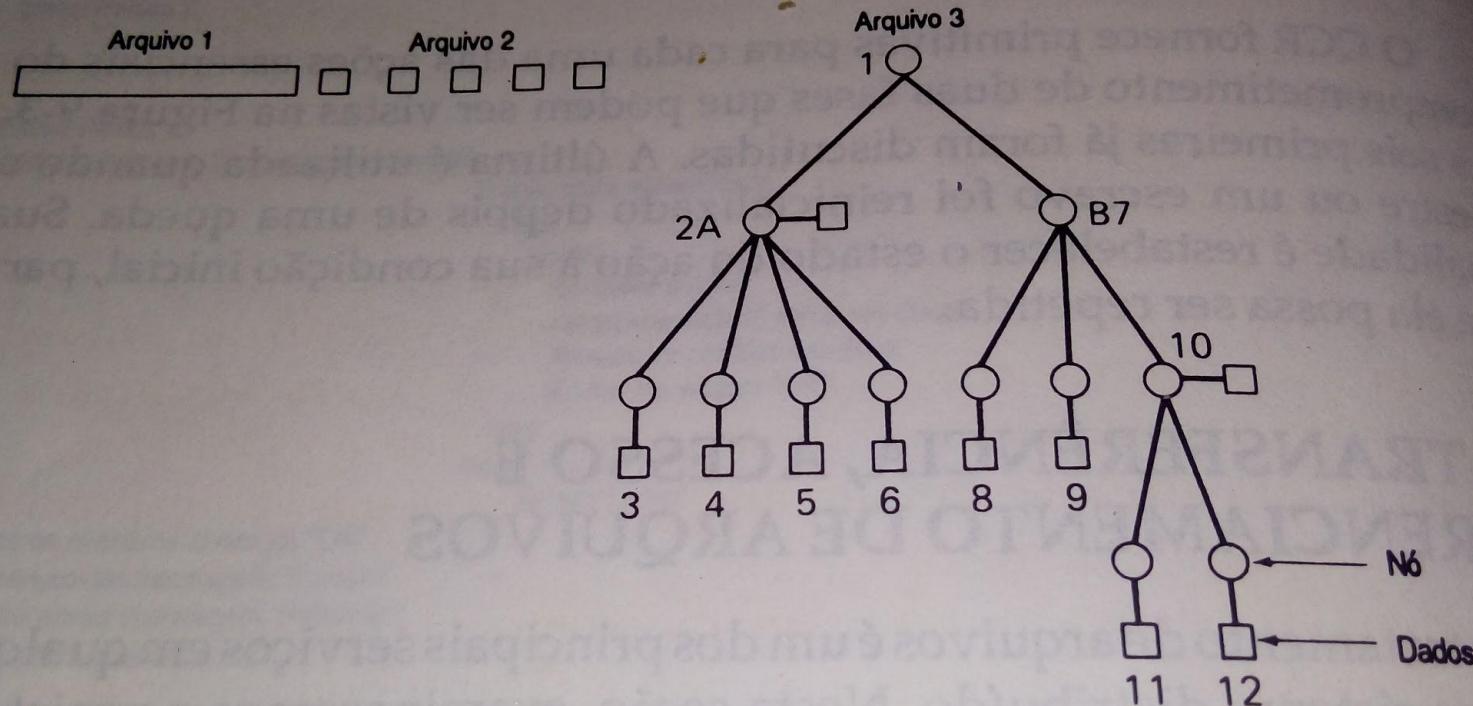


Figura 9-4 Arquivos (a) Não-estruturado (b) Sequência de registros (c)
Hierárquico

Controle de concorrência - 9.2.2

Transações são ações atômicas(ou seja indivisível), sem interferência de outros pedidos de clientes.

exemplo

comando início de transação.

abertura de arquivo e operações de arquivo.

comando finaliza a transação.

Aborta ou interromper a transação irá causar restauração do estado inicial antes da transação

Controle de Concorrência - 9.2.2

Basicamente a transação faz uma cópia de cada arquivo, o arquivo é aberto para escrita, as alterações são feitas na cópia.

se ela completar com sucesso, cópias substituem os originais

Controle de Concorrência - 9.2.2

Se dois clientes usarem o mesmo arquivo de transações simultâneas, o primeiro que encerra vence.

um exemplo compra online.

quando a última unidade de um produto é vendido, e um cliente A e outro Cliente B, estavam com carrinho do mesmo produto x, mas o cliente B comprou antes do A. então quando o A for finalizar a compra vai constar que o produto acabou, e a transação será abortada.

Controle de concorrência - 9.2.2

Observações

em transações, os arquivos vão ser sempre **imutáveis** e nunca vão sofrer alterações, por causa que sempre é uma cópia que é trocada quando tem uma transação bem sucedida..

Arquivos Duplicados - 9.2.3

Até agora vimos um servidor e vários clientes, agora vamos ver um cliente e vários servidores.

as principais funcionalidades são

- dividir a carga de trabalho pelos vários servidores.
- para permitir acesso aos arquivos mesmo se um servidor estiver inativo.
- ter confiabilidade, por ter backups independentes de cada arquivo.

Arquivos Duplicados - 9.2.3

- quando tem uma cópia de arquivo em cada n servidores, é fácil controlar o arquivo de leitura, mas quando tem uma alteração, já tem um problema por que tem que alterar todos n servidores, sobre a alteração que teve no arquivo.
- uma solução simples é chamar um servidor de principal, onde são feitas as alterações nele e propagar para os n servidores restantes.
- esse método é o mestre-escravo, onde o mestre é o principal e os n restantes servidores são os escravos
- ele é simples mas tem problema se o mestre estiver indisponível(problemas de pane ou partição de rede)

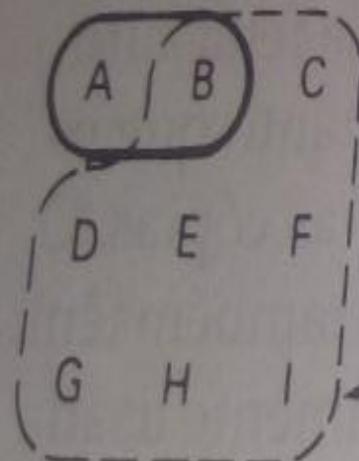
Arquivos Duplicados - 9.2.3

método de votação

ele é muito eficaz quando tem muitos clientes ativos. Seja N , o número de Servidores que contem as cópias de um arquivo.

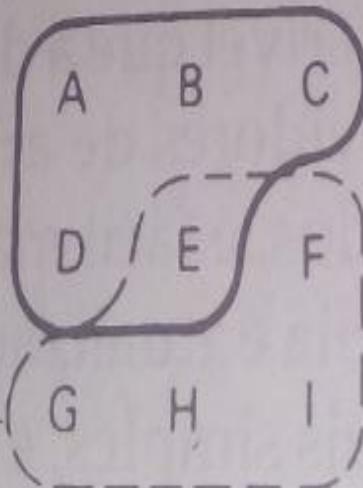
- para ler um arquivo quórum de leitura, N_l
- para alterar um arquivo quorum de escrita, N_g
- com restrição $N_l + N_g > N$.
- Normalmente N_l é menor que N_g , isso significa que a leitura vai ser mais usada, e a escrita menos.
- existe uma melhoria chamado fantasma, que diminui a ociosidade nos servidores de escrita.

$$N_r = 2, N_w = 8$$



(a)

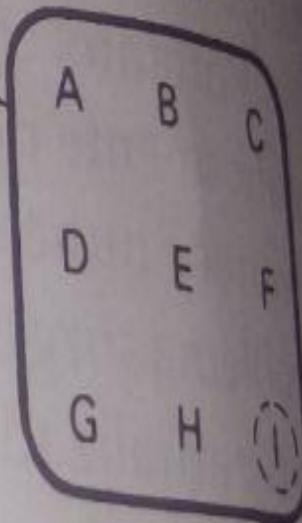
$$N_r = 5, N_w = 5$$



(b)

Quorum
de leitura

$$N_r = 9, N_w = 1$$



(c)

Figura 9-7 Três exemplos de um quorum de leitura e um quorum de escrita

Questões de Implementação - 9.2.4

A questão de implementação tem a ver como vai ser tratado o acesso ao disco e a maneira que vai ser estruturada as informações no disco.

implementação Simples

begin{Loop principal do servidor de arquivo}

 RecebeMensagem(buffer);

 FazTrabalho(buffer, resultado);

 EnviarResposta(resultado);

end

muitos acessos ao disco por um pedido pode levar a deixar o servidor ocioso enquanto espera.

Questões de Implementação - 9.2.4

Implementação por tarefa

o servidor é dividido em diversas tarefas que compartilham um espaço de endereço comum(tabelas e buffer do sistema de arquivos).

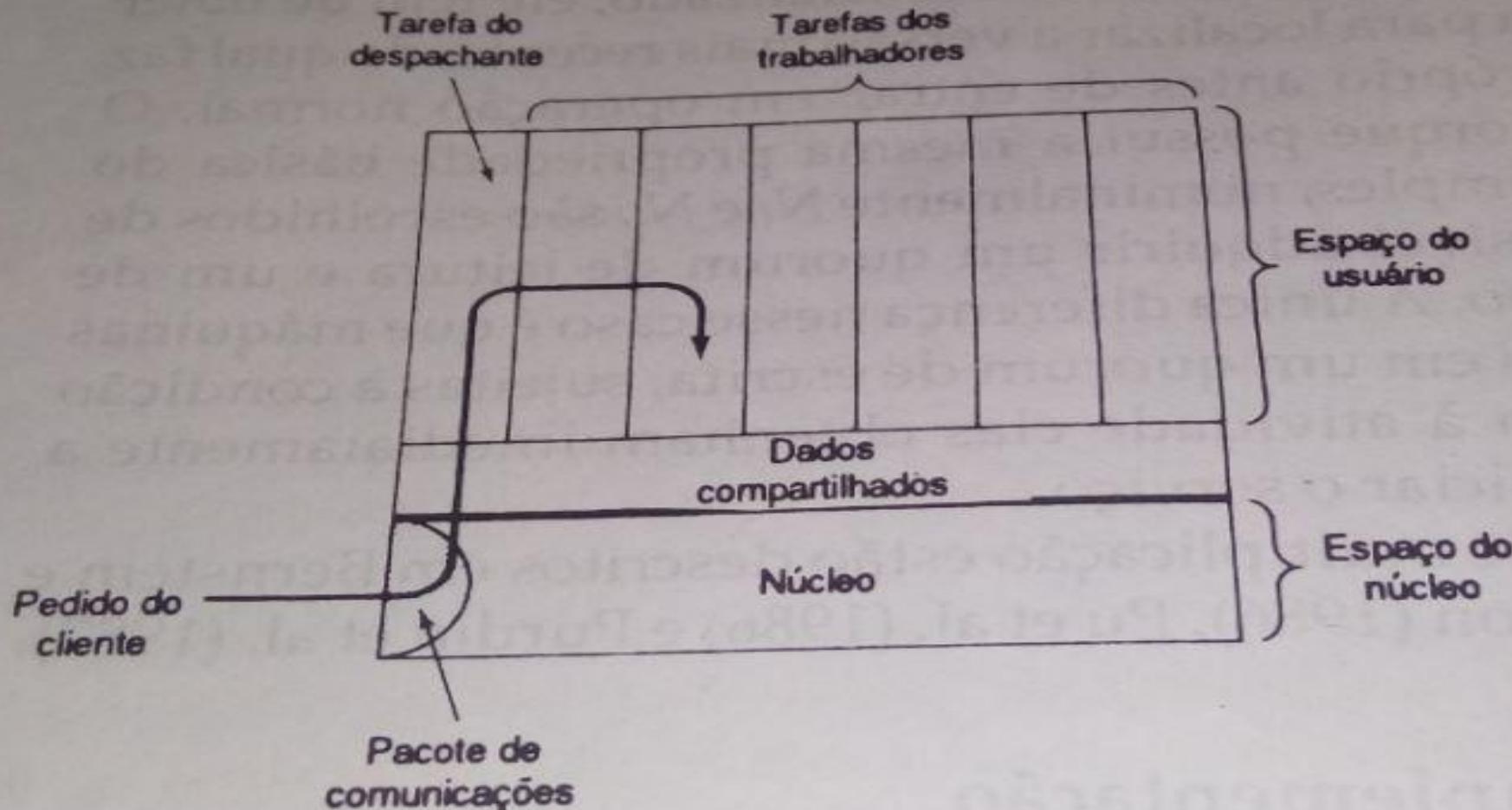


Figura 9-9 Um servidor com múltiplas seqüências

Questões de Implementação - 9.2.4

uma mensagem de solicitação é entregue ao núcleo.

O núcleo a aceita e a entrega a tarefa despachante(normalmente na espera).

é examinado e entregue a tarefa trabalhadora, é passado o ponteiro pois estão no mesmo espaço.

é executada uma tarefa por vez, se uma tarefa bloquear enquanto espera pelo disco, outra tarefa pode entrar em execução. assim a CPU não ficará ociosa.

Questões de Implementação - 9.2.4

Implementação baseada em Bloco

O disco do Servidor é dividido em blocos, que significa que, é feita uma divisão do tamanho do disco pelo número de bytes.

10 000 bytes /200 de bytes, 50 blocos

servidores baseado em bloco tentam manter os arquivos no mesmo cilindro, para não ter muita movimentação do braço do cilindro.

a técnica mais usada é manter um cache de memória de buffers mais usados(MRU).

Questões de Implementação - 9.2.4

Amoeba

- Ele armazena cada arquivo em posições contínuas no disco e o cache também é assim.
- A transferência de arquivo entre o disco e a memória pode ser feito em uma única operação de disco.
- Ele opera muito próximo da banda passante completa da LAN básica.

Questões de Implementação - 9.2.4

Servidor de arquivos sem estados

para ler e escrever um arquivo um cliente, um cliente envia um pedido especificando o arquivo(posição, chave do registro, quantidade a ser transferido)

O arquivo não precisa ser aberto e nem fechado.

Questões de Implementação - 9.2.4

Servidor com estados

O arquivo precisa ser aberto, o servidor precisa de entrada na tabela para o arquivo recém aberto.

O índice dessa entrada é retornado ao cliente para uso em pedidos subsequentes.

Questões de Implementação - 9.2.4

	vantagem	desvantagens
com estados	<ul style="list-style-type: none">• bloqueio de estados.• acessos a arquivos bloqueados.	<p>quedas de servidor, fechando todas conexões.</p> <p>responsabilidade do cliente.</p>
sem estados	<ul style="list-style-type: none">• muito resistente.• sem falhas em quedas de sistema.• fácil retorno.• não tem informação de estado para perder.	<ul style="list-style-type: none">• difícil ser implementado.• a semântica do sistema de arquivo é complicada(devido a maneira centralizada, que é muito usada).

9.3 - Correio Eletrônico (e-mail)

- Começo : Protocolo de transferência de arquivos, que por convenção, em que a primeira linha de cada mensagem continha o endereço do destinatário.
- Limitações :
 - 1 - Era inconveniente mandar uma mensagem para vários destinatários;
 - 2 - Dificuldade de processamento, devido a falta de estrutura;
 - 3 - O remetente não sabia se a mensagem era ou não recebida;
 - 4 - Não era possível enviar mensagens contendo uma mistura de texto, voz e imagem.

9.3.1 - Arquitetura e Serviços MOTIS e X.400

- 1984 : CCITT -> X.400.
- X.400 base para o MOTIS (ISO 10021).
- Preocupação com todos os aspectos do sistema de e-mail, desde quando o destinatário resolve escrever a mensagem até quando o remetente a exclui.
- Possui 6 características básicas :

1 - Composição : processo de criação da mensagem, cuida do endereçamento e do cabeçalho;

2 - Transferência : movimentação da mensagem do remetente para o destinatário, deve acontecer automaticamente;

3 - Relatório : diz ao remetente o que ocorreu com a mensagem;

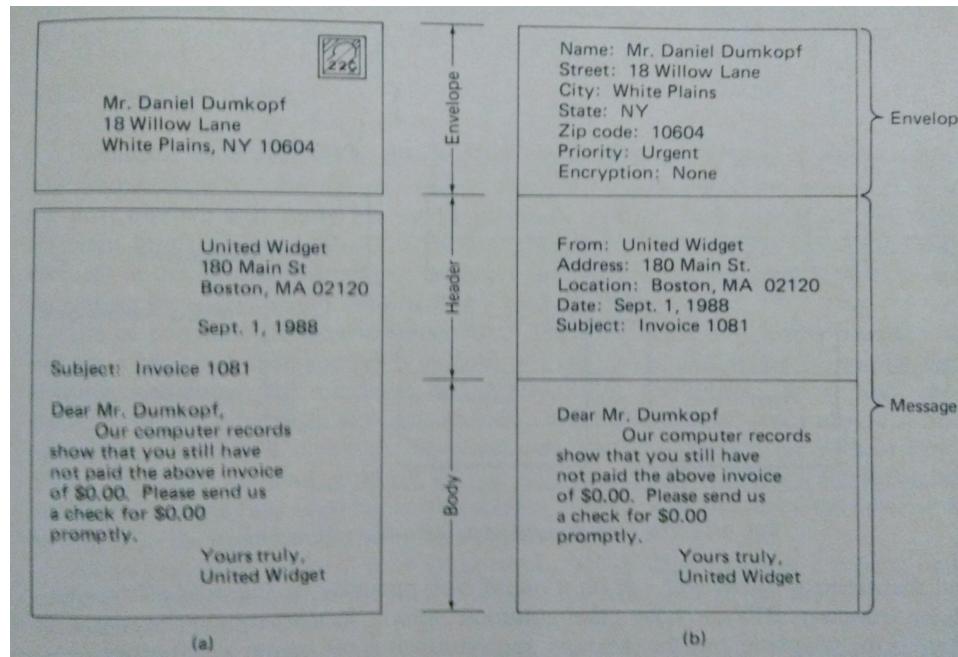
4 - Conversão : possibilita que a mensagem seja lida em diferentes dispositivos;

5 - Formatação : estrutura usada para mostrar a mensagem no terminal do destinatário;

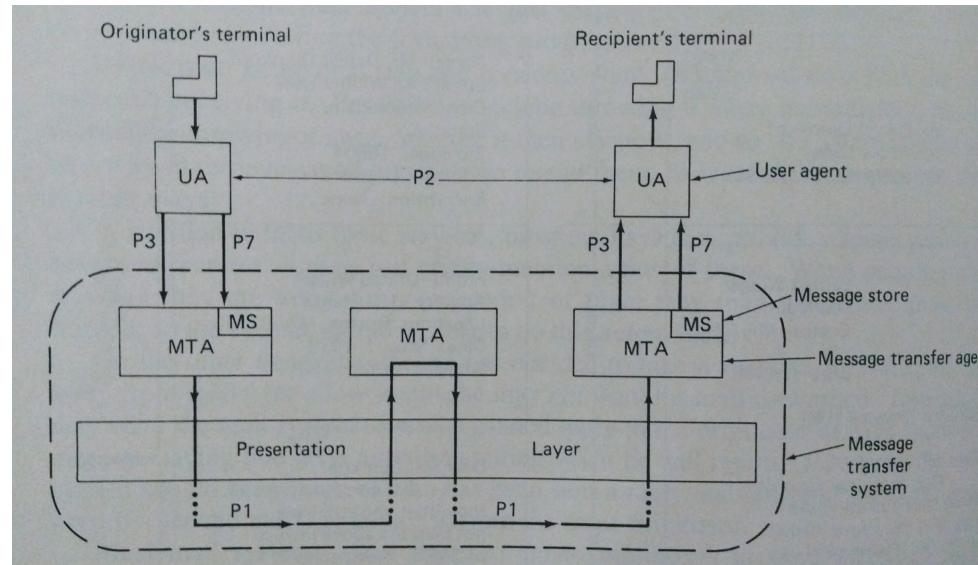
6 - Disposition : o que o destinatário faz após receber a mensagem;

- Além dos serviços básicos o sistema pode prover algumas funções mais avançadas como :
 - Encaminhamento de mensagens;
 - Canned reply, avisa o remetente sobre indisponibilidade;
 - Caixa de entrada, armazenamento de mensagens;
 - Distribution list, lista de endereços de e-mail;
 - Carbon copies;
 - Prioridade;
 - Criptografia.

- Distinção entre envelope e conteúdo.
- Envelope : encapsula a mensagem, contém os parâmetros necessários para o transporte e interpretação da mensagem.



- Três tipos de mensagens : user messages, replies e probes.
- User messages : contém informações enviadas entre usuários.
- Replies : reportam ao remetente se sua mensagem foi ou não recebida.
- Probes : mensagens com envelope vazio, usada para verificar se o destinatário é atingível, e evidenciar rota tomada pela mensagem (selo de tempo e localização).



- User Agent : programa que provê uma interface com o sistema de e-mail.
- Message Transfer Agent : aceita mensagens de user agents, funciona como o correio.
- É possível ter user e message transfer agent na mesma máquina, porém a melhor forma é ter o primeiro rodando em um computador pessoal e o segundo em um mainframe.
- Problema : o que fazer com mensagens longas ou mensagens que chegam quando o usuário não está logado.
- Solução : message transfer agent ter uma área de armazenamento para cada usuário, mensagens são mantidas nelas até que o usuário logue.
- Message transfer system : conjunto de todos message transfer agents.
- Interpersonal messaging (IPM) : protocolo usado para comunicação entre user agents.

9.3.2 - User Agent

- Comunicação entre terminais dos usuário.
- Comunicação com message transfer agent em relação a entrega e envio de mensagens.
- Lida com o armazenamento de mensagens.
- Invocado ao executar um programa que aceita comandos relacionados a composição, envio e recepção de mensagens.
- Normalmente o programa vai procurar por mensagens e as exibi-las para o usuário, são mostrados campos do envelope da mensagem, os campos exibidos podem ser escolhidos pelo usuário ou não, de modo a mostrar apenas aqueles determinados pelo programa.

Command	Parameter	Description
h	#	Display header(s) on screen
c		Display current header on screen
t	#	Type message(s) on screen
s	address	Send a message
f	#	Forward message(s)
a	#	Answer message(s)
d	#	Delete message(s)
u	#	Undelete previously deleted message(s)
m	#	Move message(s) to another mailbox
k	#	Keep message(s) after exiting
r	mailbox	Read a new mailbox
n		Go to next message and display it
b		Back up to previous message and display it
g	#	Go to a specific message but do not display it
e		Exit mail system and update mailbox

- User Agents sofisticados possibilitam lidar com grandes volumes de mensagens.
- O protocolo usado pelos user agents basicamente remete ao uso dos headers de cada mensagem.

Field	Description
Originator	Who actually sent the message
Authorizing users	On whose behalf was it sent
Primary recipients	To whom is the message addressed
Copy recipients	Who gets carbon copies
Blind copy recipients	Who gets carbon copies secretly
Reply recipients	To whom should the reply be sent
Reply time	By when the reply desired
Message id	Message identifier
In reply to	Message to which this is a reply
Obsoleted messages	Messages invalidated by this one
Related messages	Other messages relevant to this one
Subject	What is the message about
Importance	Message priority
Sensitivity	Public, company confidential, etc.
Expiry time	Time when message ceases to be valid

- O conteúdo da mensagem consiste em um estrutura simples, sendo dividida em parte principal, opcional e um ou vários anexos.

9.3.3 - Message Transfer Agent

- Retransmite a mensagem do remetente para um destinatário.
- Processo é diferente de configurar uma application-layer association, que é feito em FTAM.
- FTAM, impossibilidade de ler um arquivo remoto se o servidor está offline.
- O serviço ocorre de modo, store-and-forward, hop-by-hop .
- Processamento de mensagens :
 - Se veio de um user agent, sintaxe é verificada, se inválida é mandada de volta com uma explicação;
 - Caso contrário, são fixados um identificador e um selo de tempo e passa a ser tratada como se tivesse vindo de outro message transfer agent;
 - Verifica se o user agent do destinatário ou sua caixa de entrada é local, se sim a mensagem pode ser entregue, enfileirada para entrega e armazenada na caixa de entrada;
 - Caso contrário, a mensagem é encaminhada para outro message transfer agent;
 - Se necessário uma confirmação é enviada ao remetente.

- Entregas a user agents locais nem sempre são triviais devido ao uso de equipamentos diferentes, message agent pode tentar converter antes de entregar a mensagem, porém nem todas conversões são possíveis.
- Remote operation service :
 - Invocar operações em outro computador;
 - Retornar o resultado da operação remota;
 - Retornar mensagem de erro;
 - Rejeita operação remota.
- Pode ser usado que um message transfer executar uma operação de entrada de mensagem em outro.

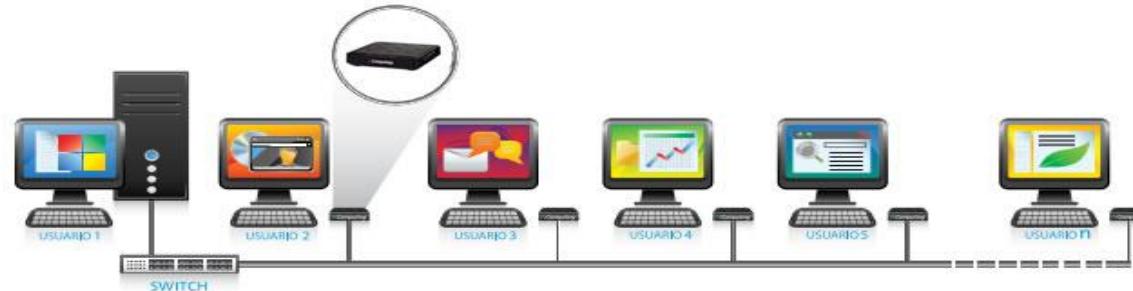
Field	Description
Originator's address	Mail address of the sender
Recipient's address	Mail address of the recipient
Alternate recipient allowed	Is redirection to someone else allowed
Alternate recipient	Second choice recipient
Message id	Message identification
Priority	Slow (cheap), normal, fast (expensive)
Originator report request	What reporting does originator want
MTA report request	What reporting does MTA want
Deferred delivery	Do not deliver before this time
Latest delivery	Do not deliver after this time
Content return	Should content be returned on nondelivery
Information type	Text, facsimile, digitized voice, etc.
Conversion prohibited	No conversion allowed
Lossy conversion prohibited	Only perfect conversion allowed
Explicit conversion	Conversion is known to be required
Encryption identification	Index into encryption key table
Content integrity check	Check sum on contents
Originator's signature	Digital signature
Message security label	Classified, secret, top secret, etc.
Proof of delivery	Recipient's signature

9.4 Terminais Virtuais

Os terminais virtuais funcionam como uma CPU porém não armazenam informação local, o sistema operacional e os demais softwares são centralizados no servidor.

Três classes principais:

- Modo de rolagem (mais simples)
- Modo de página
- Modo de formulário (mais sofisticado)



9.4.1 Terminais de Modo de Rolagem

- Não possuem um microprocessador integrado nem a capacidade de edição
- Quando uma tecla é pressionada o caracter é enviado pelo barramento e pode também ser exibido
- Quando um carácter chega pelo barramento ele é apenas exibido.
- A medida que novas linhas vão aparecendo as antigas são mandadas para cima (scroll upward).
- Por não possuírem poder de processamento eles não podem se comunicar pela rede usando os protocolos padrões.
- A solução é a caixa preta PAD (Packet Assembler/Disassembler) que é colocada entre o terminal e a rede

9.4.1 Terminais de Modo de Rolagem

Por serem simples, estes terminais podem ser diferentes no que tange

- Conjunto de caracteres
- Comprimento de linha
- Presença ou ausência de eco automático e a sobreimpressão
- Maneira como são tratados os retornos de cursor, avanço de página
- Tabulação horizontal e vertical

Alguns terminais também apresentam problemas potenciais de sincronização, tais como número de caracteres de preenchimento necessários em seguida a uma tabulação ou retorno de cursor.

9.4.1 Terminais de Modo de Rolagem

Sobre o PAD

- O CCITT (Comitê Consultivo Internacional Telegráfico e Telefônico) definiu interfaces padrões para o PAD, que não é exatamente um terminal virtual, não cabe muito bem na hierarquia do OSI nem em outro lugar.
- Quando um terminal estabelece contato com um PAD, o operador ou o próprio computador define parâmetros que descrevem a conversação entre o terminal e o PAD.
- Os parâmetros são numerados e também seus valores de opção. Um exemplo de comando seria SET 1:0 que significa fixar o parâmetro 1 com o valor 0.

Parâmetro	Descrição	Valores permitidos
1	O operador do terminal pode sair do modo de transferência de dados ou alterar os parâmetros do PAD	0 = Não (escape proibido) 1 = Sim (escape permitido)
2	Os caracteres de eco do PAD devem retornar ao terminal ?	0 = Não 1 = Sim
3	Que caracteres devem disparar o PAD no envio de um pacote parcialmente completo ?	0 = Somente enviar pacotes completos 1 = Retorno de cursor 126 = Todos os caracteres de controle + DEL
4	Com que velocidade deve o PAD se interromper e enviar um pacote parcialmente completo ?	0 = Sem interrupção 1-255 = Tempo em tiques de 50ms
5	Pode o PAD proibir (temporariamente) o terminal de enviar caracteres ?	0 = Não (modo simples) 1 = Sim (controle de fluxo)
6	É permitido ao PAD enviar sinais de serviço para o terminal ?	0 = Não (suprimir sinais) 1 = Sim (entregar sinais) 0 = Nada 1 = Interrromper
7	O que deve fazer o PAD ao receber um sinal de interrupção do terminal ?	2 = Reinicializar 4 = Enviar ao host um pacote de controle 8 = Sair para o modo de comandos 16 = Descartar a saída
8	Deve o PAD descartar a saída de computador destinada ao terminal ?	0 = Não (Entregar) 1 = Sim (Descartar)
9	Quantos caracteres de preenchimento o PAD deve inserir depois de dar saída a um retorno de cursor para o terminal ?	0 = Nenhum 1-7 = Número de fillers

10	O PAD deve empacotar a saída para evitar estouro na linha ?	0 = Não 1 = 255 (sim) tamanho da linha 0 = 110 8 = 200
11	Velocidade do terminal (bps)	1 = 134,5 9 = 100 2 = 300 10 = 50
12	O terminal pode proibir (temporariamente) o PAD de lhe enviar alguma saída ?	0 = Não 1 = Sim
13	Deve o PAD inserir um avanço de linha depois do retorno de cursor ?	0 = Não 1-7 Condicoes diversas
14	Deve o PAD acrescentar um filler depois dos avanços de linha ?	0 = Não 1-7 = Sim (Quantos)
15	Deve o PAD permitir a edição ?	0 = Não 1 = Sim
16	Seleciona o que deve ser digitado para eliminar um caráter	0-127 = caráter
17	Seleciona o que deve ser digitado para eliminar uma linha	0-127 = caráter
18	Seleciona o que deve ser digitado para exibir uma linha	0-127 = caráter

Figura 9-16 Os parâmetros do PAD (do padrão X.3 do CCITT)

9.4.2 Terminais de Modo de Página

- Geralmente são terminais que mostram 25 linhas de 80 caracteres cada.
- Pode-se movimentar o cursor do mouse pela tela para modificar parte do que é mostrado.
- Possui os mesmos problemas do terminal de modo de rolagem e mais alguns, tais como: tamanho da página, o endereçamento do cursor, presença ou ausência de piscamento, vídeo reverso, cor e intensidades múltiplas.
- Não existem dois terminais diferentes que utilizem os mesmos comandos para a realização das funcionalidades. Isso torna difícil escrever editores de tela e outros tipos de softwares baseados no vídeo.

9.4.2 Terminais de Modo de Página

- Uma solução amplamente utilizada, cujo pioneirismo se deve ao Berkeley UNIX, é definir um terminal virtual constituído pelos comandos apresentados pela maioria dos terminais de modo de página.
- Quando um editor se inicia, ele pede o tipo de terminal e depois lê a entrada para o terminal de um banco de dados chamado *termcap* (característica do terminal). Essa entrada determina a sequência de escape requerida para cada comando virtual.
- Desde que o software se limite a transmitir comandos do terminal virtual, ele irá funcionar em qualquer terminal que tenha uma entrada *termcap*.

Código	Tipo	Descrição
bs	Booleano	O terminal pode usar CTRL-H como backspace
hc	Booleano	O terminal imprime cópia em papel
nc	Booleano	O terminal é um CRT mas não é capaz de rolagem
os	Booleano	O terminal pode fazer superposição (múltiplos caracteres na mesma posição)
ul	Booleano	O terminal pode sublinhar texto
pt	Booleano	O terminal possui tabulações por hardware
co	Inteiro	Número de colunas em uma linha
li	Inteiro	Número de linhas na tela ou em uma página
dB	Inteiro	Número de milissegundos de retardo necessários para o backspace
dC	Inteiro	Número de milissegundos de retardo necessários para retorno de cursor
dN	Inteiro	Número de milissegundos de retardo necessários para avanço de linha
dT	Inteiro	Número de milissegundos de retardo necessários para tabulação
cd	String	Limpa a tela da posição do cursor até o fim do vídeo
ce	String	Limpa a tela da posição do cursor até o fim da linha
cl	String	Limpa toda a tela
cm	String	Movimento do cursor
ct	String	Limpa todos os pontos de tabulação
dc	String	Elimina o caráter sob o cursor
dl	String	Elimina a linha que contém o cursor
ff	String	Vai para o topo da página seguinte (terminais de hardcopy)
ic	String	Insere um caráter na posição do cursor
is	String	String usado para inicializar o terminal (e talvez fixar tabulações)
mb	String	Entra no modo de piscamento
md	String	Entra no modo bold (brilho extra)
us	String	Entra no modo de sublinhado
mh	String	Entra no modo de dim (metade do brilho)
so	String	Entra no modo de vídeo reverso
me	String	Entra no modo normal (nenhum piscamento, bold, dim, vídeo reverso, etc.)
pc	String	Caráter a usar para preenchimento depois de tabulação, avanço de linha, etc.
sf	String	Rola a tela para a frente
sr	String	Rola a tela para trás
kl	String	Seqüência de escape enviada pela tecla da seta esquerda
kr	String	Seqüência de escape enviada pela tecla da seta direita
ku	String	Seqüência de escape enviada pela tecla da seta para cima
kd	String	Seqüência de escape enviada pela tecla da seta para baixo
kh	String	Seqüência de escape enviada pela tecla home
kb	String	Seqüência de escape enviada pela tecla backspace

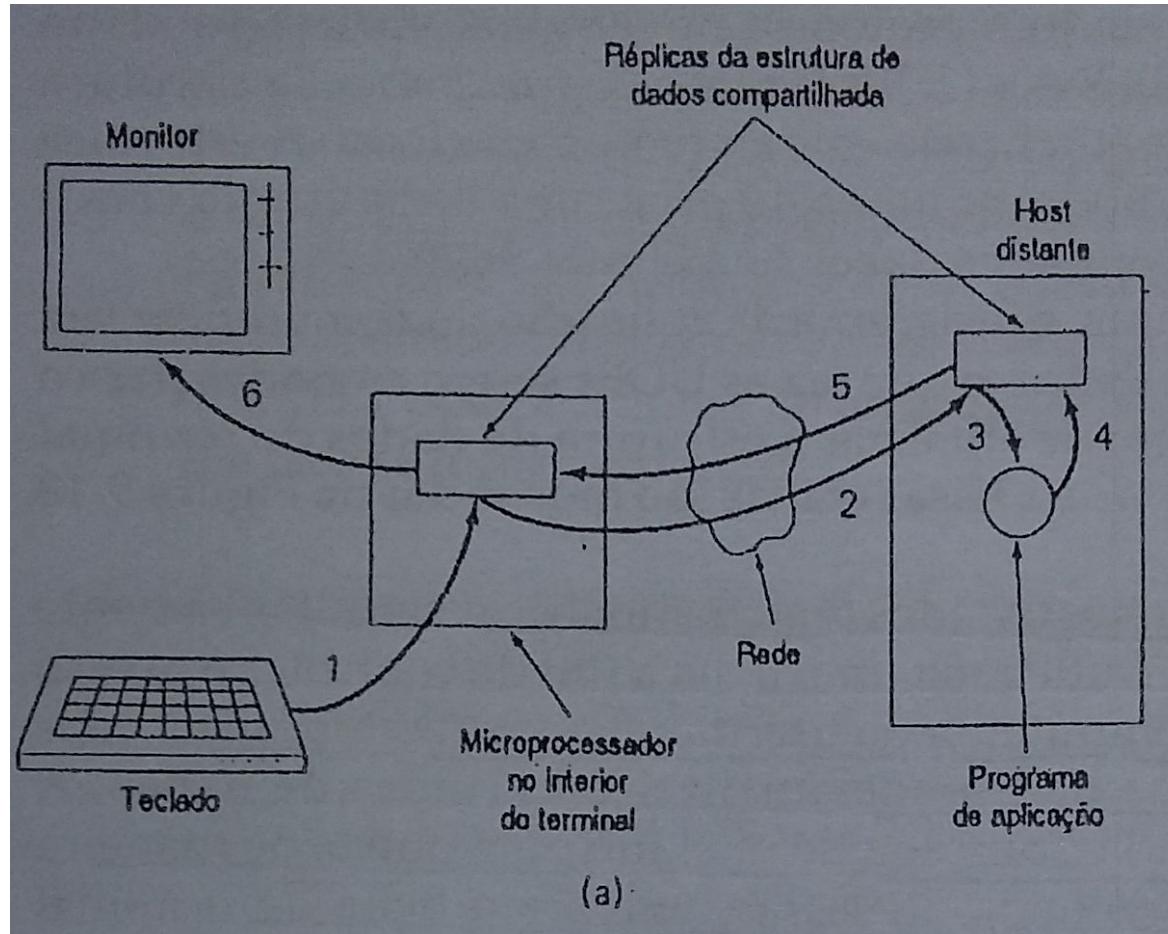
Figura 9-17 Algumas das entradas mais comuns no banco de dados de termcap

9.4.3 Terminais de Modo de Formulário

- Possui microprocessador embutido.
- O computador baixa um formulário, em que alguns dos campos são apenas para leitura contendo informações e outros que podem ser preenchidos.
- O microprocessador possibilita edição local, uso de macros e outras funcionalidades.
- Quando um formulário é preenchido o microprocessador pode fazer uma rápida checagem de sintaxe. Se tudo está correto a parte modificada está pronta para ser posta na rede para chegar ao computador.

9.4.3 Terminais de Modo de Formulário

- Pode possuir um modelo que consiste de estrutura de dados compartilhadas. Nesse modelo o software do terminal virtual mantém um abstração do display mostrado. O terminal virtual tem a responsabilidade de alterar a tela quando há alteração na abstração.
- O modelo anterior possui duas variantes, a síncrona e a assíncrona.
- O coração da representação abstrata é um array de caracteres que representam o conteúdo da tela.
- Cada elemento do array possui uma coordenada exclusiva e contém apenas um caracter do alfabeto.



(a)

9.4.3 Terminais de Modo de Formulário

- Nesse modelo, existe uma única estrutura de dados abstrata representando a imagem na tela. Duas cópias idênticas dessa estrutura são mantidas, uma pelo software do terminal virtual em execução no microprocessador e a outra pelo software do terminal virtual em execução próximo ao programa de aplicação em computador host distante.
- A pessoa no terminal pode modificar a cópia da estrutura de dados abstrata no microprocessador digitando no teclado. Essas alterações são visíveis na tela porque a exibição em tela é controlada pela estrutura de dados local.

9.4.3 Terminais de Modo de Formulário

- O microprocessador transmite comandos do terminal virtual para o host distante pela rede, usando o protocolo de terminal virtual (rotulado como 2). Essas UDPs fazem a cópia remota da estrutura de dados ficar sempre atualizada em relação à cópia local. A estrutura de dados modificada pode ser lida pelo programa de aplicação com o uso de comandos apropriados (label 3).
- De modo semelhante, o programa de aplicação pode modificar sua cópia de estrutura de dados, o que faz as UDPs serem enviadas para o microprocessador a fim de atualizar a estrutura de dados do terminal e portanto sua tela de vídeo (etapas 4,5,6).

9.4.3 Terminais de Modo de Formulário

- Para evitar que ambos os lados tentem modificar simultaneamente a estrutura de dados, é utilizado um *token* a fim de controlar o acesso para a atualização.
- O computador ou o terminal podem reter o *token*, mas somente o detentor pode atualizar a estrutura de dados. O *token* pode ser solicitado e passado de um lado para outro, de maneira análoga aos *tokens* da camada de sessão.

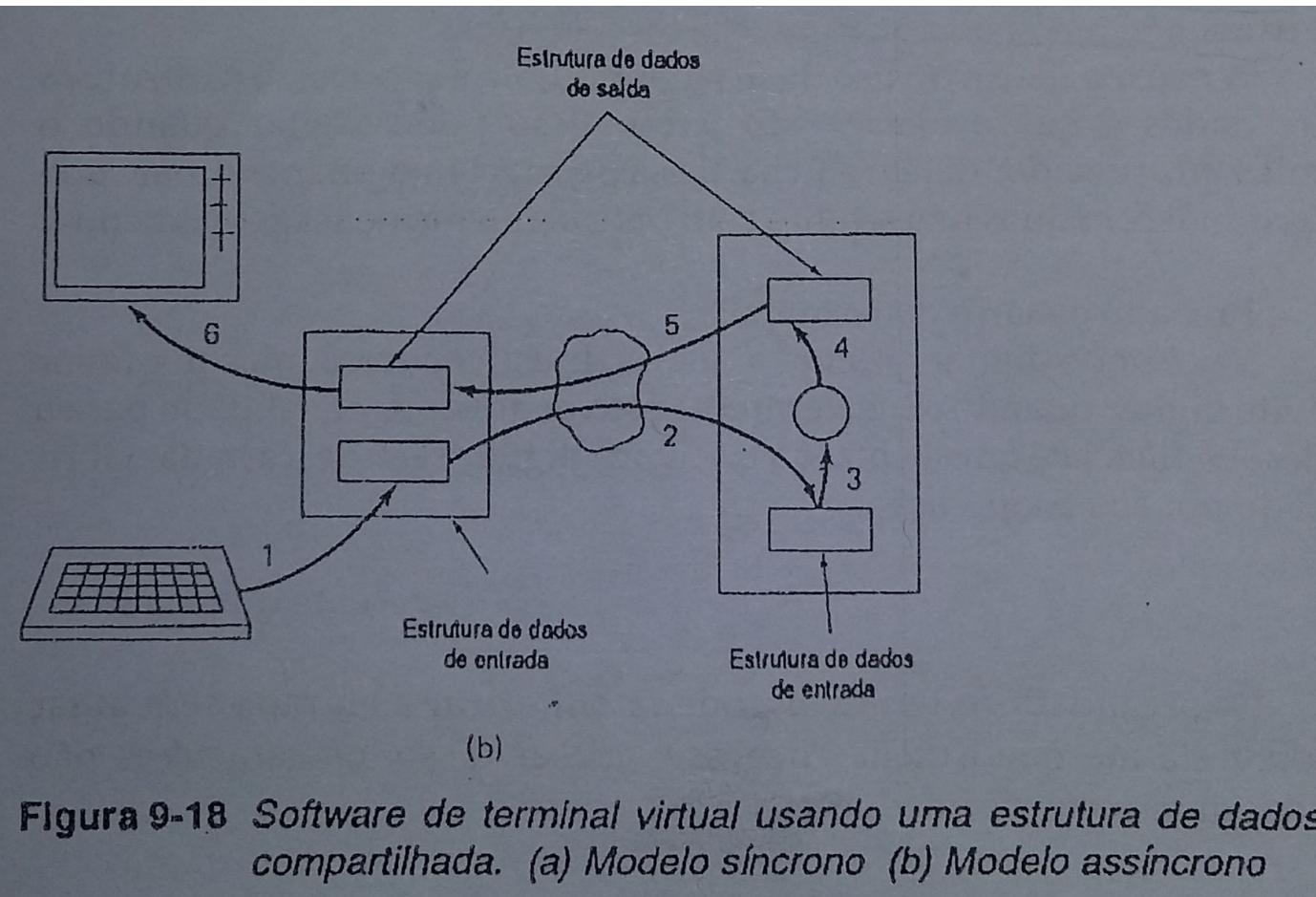


Figura 9-18 Software de terminal virtual usando uma estrutura de dados compartilhada. (a) Modelo síncrono (b) Modelo assíncrono

9.4.3 Terminais de Modo de Formulário

- O modelo assíncrono, consiste em dois monólogos independentes, em lugar de um único diálogo. Cada extremidade da conexão tem uma estrutura de dados para entrada e uma segunda para saída. A alteração da sua própria estrutura de dados de saída afeta a estrutura de dados de entrada de seu parceiro, mas não afeta a sua estrutura de dados de entrada ou a estrutura de dados de saída do parceiro.
- No modelo assíncrono, cada cópia da estrutura de dados possui um único leitor e um único escritor e assim não pode ocorrer conflitos de escrita. Porém o preço pago por isso é o armazenamento adicional.

9.4.3 Terminais de Modo de Formulário

- A parte central da estrutura de dados abstrata é um array de caracteres representando o conteúdo da tela. O array pode ser: unidimensional (para modelar as máquinas de escrever inteligentes com uma linha de texto que pode ser editada), bidimensional (para modelar terminais de CRT simples) ou tridimensional (para modelar terminais CRT com múltiplas páginas de texto ou múltiplas janelas). As coordenadas(x,y,z) indicam posição na tela.
- Proporcionam suporte direto a formulários.
- Efeito Líquido.
- Controle de Entrega.
- Tratamento da atenção.

9.5 - Outras Aplicações

9.5.1 -> Serviço de Diretório

- Services, SAPs, telefones, entre outras entidades ligadas em rede sempre possuem um número único que à identifica
- Problema : Pessoas tendem a não lembrar o numero , alem disso o número teria que ser fixo.
- Mapeamento de nomes em números
- Necessidade de um esquema geral

9.5.1 - Serviço de Diretórios

- X.500 - Série de padrões para redes de computadores abordando serviço de diretórios , a série X.500 foi desenvolvida pelo ITU-T.
- Os serviços de diretório foram desenvolvidos para dar suporte aos requisitos do padrão X.400.
- A ISO foi parceira no desenvolvimento do padrão, incorporando-os ao pacote de protocolos OSI.

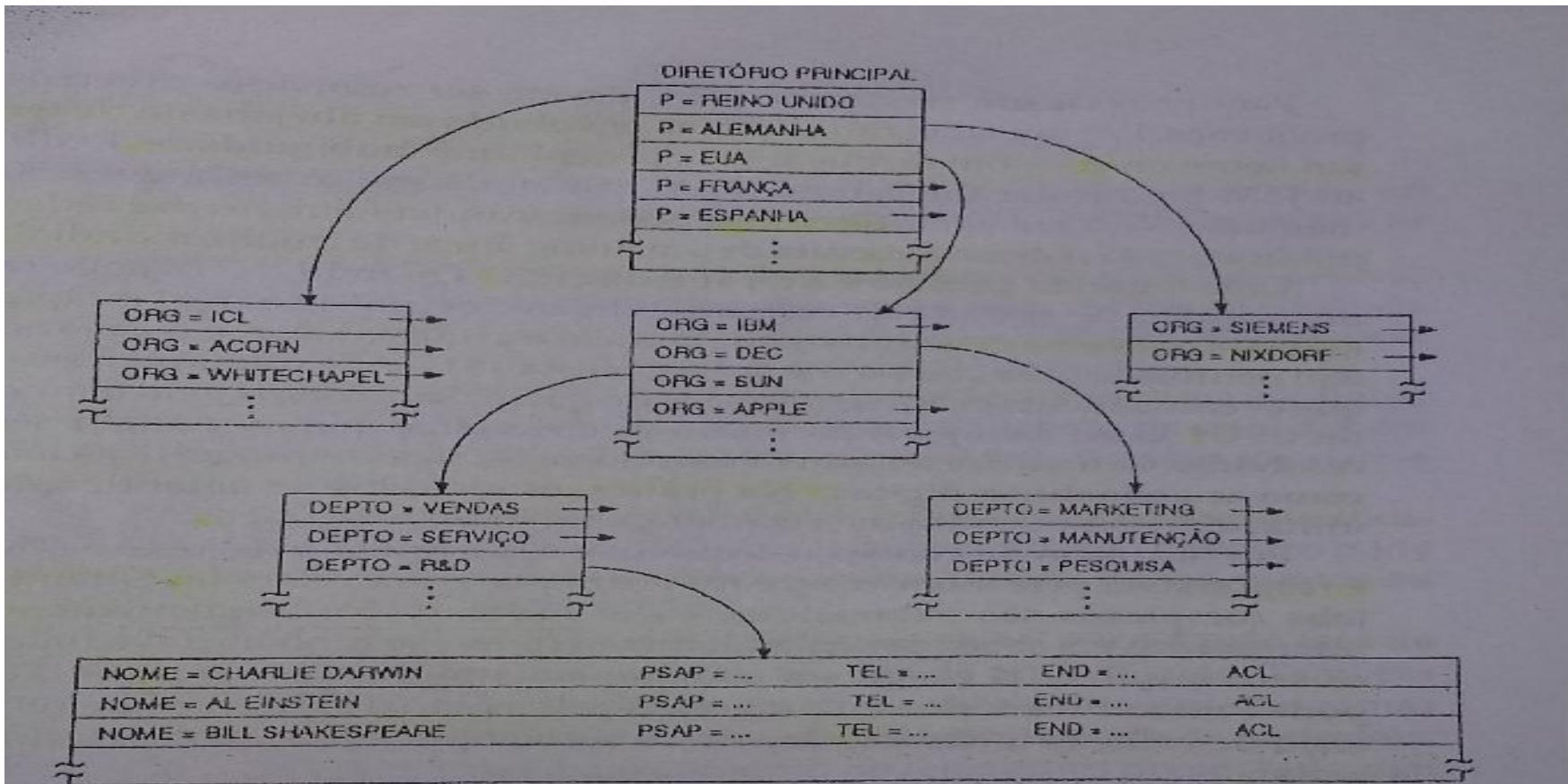
Os protocolos definidos pelo X.500:

- **DAP** ([Directory Access Protocol](#))
- **DSP** ([Directory System Protocol](#))
- **DISP** ([Directory Information Shadowing Protocol](#))
- **DOP** ([Directory Operational Bindings Management Protocol](#))

9.5.1 - Serviço de Diretórios

- O conceito primário do X.500 é que há um único Directory Information Tree (DIT), uma organização hierarquizada de entradas que são distribuídas por servidores. Uma entrada consiste de um conjunto de atributos, cada atributo com um ou mais valores. Cada entrada tem um único Distinguished name, formado pela combinação de seu Relative distinguished name (RDN), um ou mais atributos da própria entrada, e o RDN de cada entrada superior hierarquicamente até a raiz do DIT.

9.5.1- Serviço de Diretórios



9.5.2 - Transferência e Gerenciamento de Jobs (JTM)

- Aplicações que lidam com o movimento de um job, que por exemplo foi criado em um desktop para ser executado em um mainframe.
- O usuário pode especificar o que deve ser feito, onde e quais arquivos são de entrada e saída. O trabalho do JTM nesse caso deve garantir que isso ocorra.
- Existem várias formas de fazer o serviço anterior :
 - 1^a : requisitar para cada máquina que o possui.
 - 2^a : Máquina de iniciação orienta as outras.
 - 3^a : passar de máquina por máquina reunindo-os

9.5.2 - Transferência e Gerenciamento de Jobs (JTM)

- Para o JTM é desconhecida a visão do conteúdo de arquivos, as linguagens de controle de jobs ou a natureza do processamento.
- JTM também pode prover informações sobre a execução do arquivo (monitoramento), como por exemplo por quanto tempo executou.

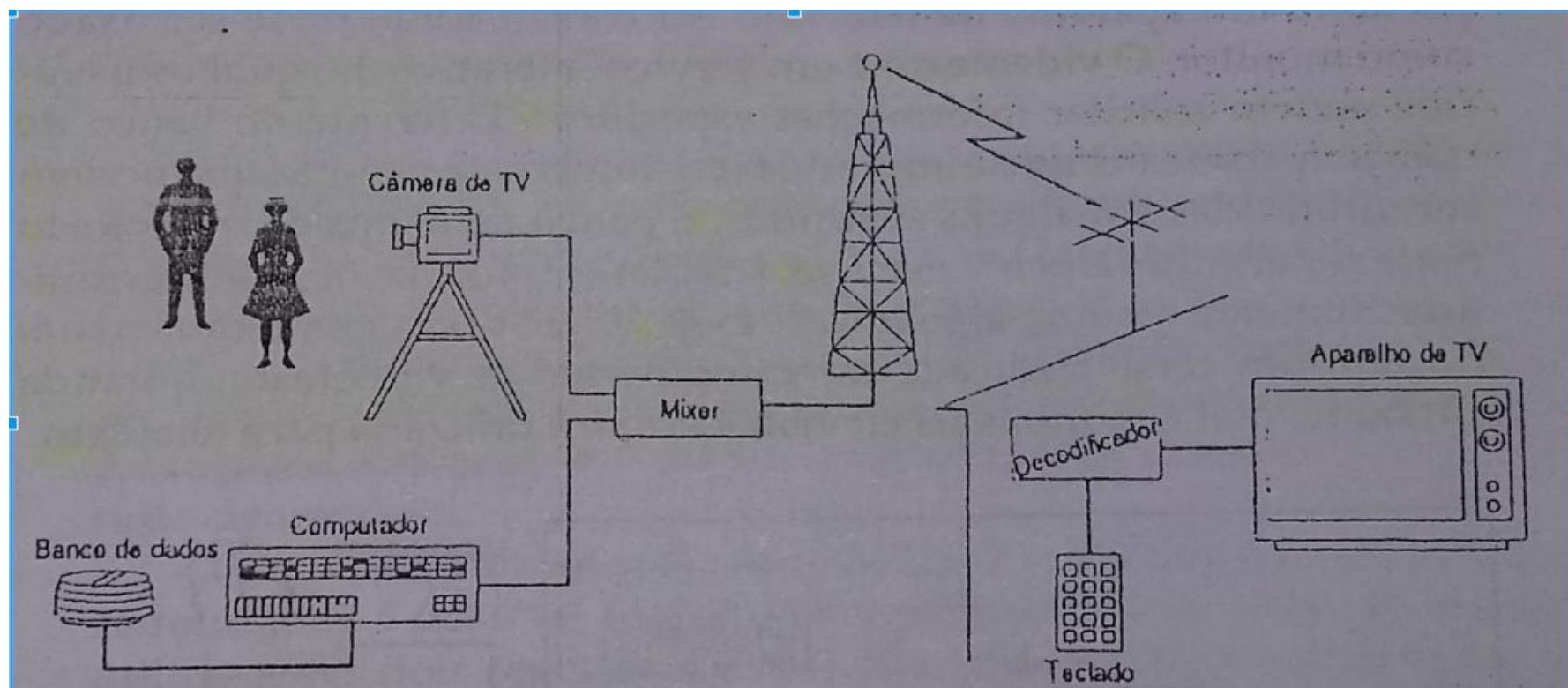
9.5.3 - Transferência e Armazenamento de Figuras

- Atenção Figuras não Fotografias
- Foi pensado devido ao aumento da resolução dos monitores que implicaria na demanda por transmissão de imagens.
- Figuras + Texto
- É possível especificar :
 - Tipo de Linha
 - Cor
 - Espessura
 - Textura
- Arquivo de figuras
- Aplicações típicas: Esquemas elétricos, plantas e etc

9.5.4 Teletext e Videotex

- Teletex : páginas de informações transmitidas via broadcast por sinais de televisão
- Inclui informações nacionais, internacionais e desportiva, previsão do tempo, programação dos canais e alguns jogos simples
- Aproveita do pequeno tempo, que leva para os elétrons da televisão se organizarem ,para o envio das páginas.

9.5.4 - Teletext e Videotex



9.5.4 - Teletext e Videotex

100B Teletext 100 Sep13 18:41:26

TELETTEXT HD

Weather from Mon 13/08 to Mon 18/08 1/3

GOLF

STENSON POWERS TO PLAYERS TITLE 481
Vaneenk holds off Daly in Italy 482
Players Championship Leaderboard 483
Scores from worldwide tournaments 484
Latest rankings & events calendar 489

CLOUDS

Cloudy with outbreaks of rain

BRIGHT SPOTS

Small risk of a shower

TELEVISION 480 **WEATHER** 102
TECH MAGAZINE 480 **TELEVISION** 102
GAMES 480 **LOTTERY** 102
ENVIRONMENT 480 **MOROSCOPES** 102
COMPUTER GAMES 480 **SPORT** 102
LIFESTYLE 480 **REGIONAL SPORT** 102
HOROSCOPES 480 **SOCCER** 102
KIDS 480 **ROADS/TRAVEL** 102
FINANCE 480 **HOLIDAYS** 102
RACING 480 **FLIGHTS** 102
CINEMA 480 **FLIGHT ARRIVALS** 102
MUSIC 480 **UK/WORLD NEWS** 102
WHAT'S ON 480 **NEWS POLL** 102
COURSES/JOB JOBS 480 **REGIONAL NEWS** 102
HOME COMPUTING 480 **LIVE QUIZZES** 102
EDUCATION 480 **CLASSIFIED ADS** 102
OVER FIFTIES 480 **DATING SERVICE** 102

FTSE100 at CLOSE Winners 66 Losers 32

FTSE	6595.80	+47.90	at CLOSE
techMARK100	1723.32	+9.40	at CLOSE
Dow Jones	14071.3	+98.98	at 18:24
BBCGlobal30	6153.33	+23.76	at 18:22
Nikkei	17065.0	-27.45	at CLOSE
DEurostoxx	4455.31	+34.92	at CLOSE

Full Index 299 Dow Jones Report 232

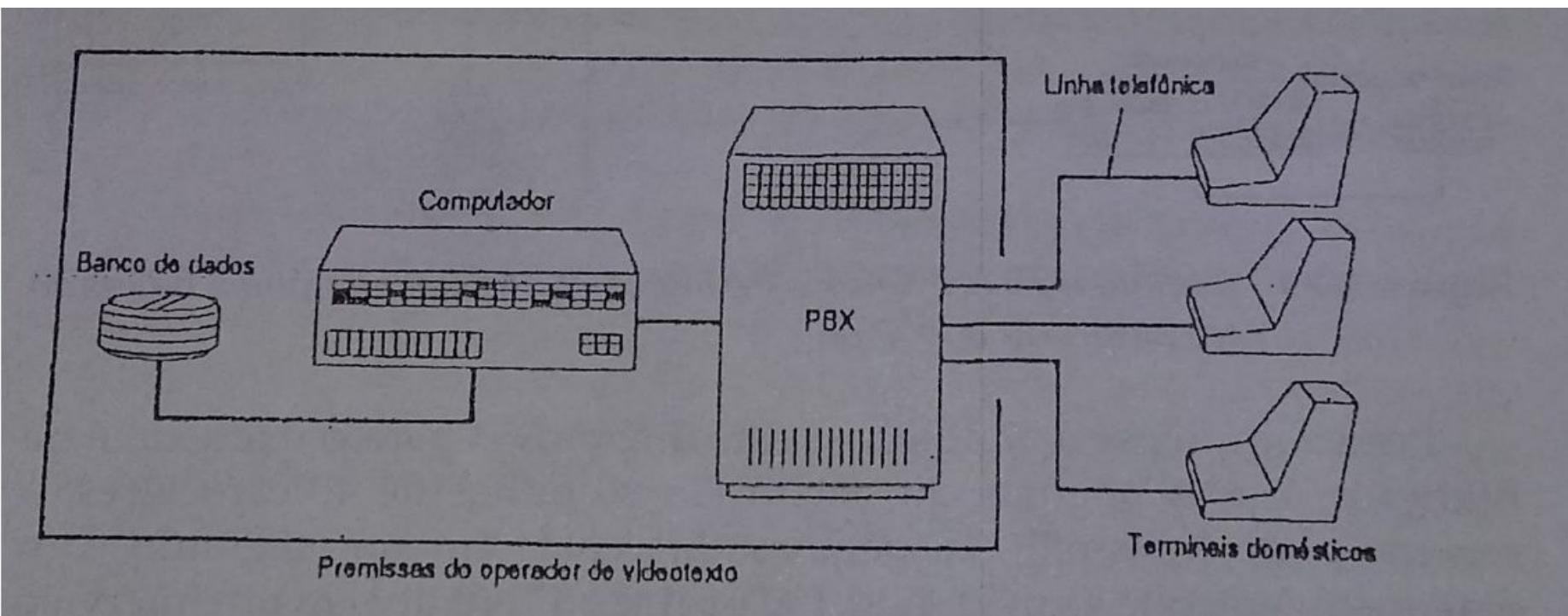
LOW RATE LOANS!!! SEE p357

TV Guide **Sport** **News** **Holidays**

9.5.4 - Teletext e Videotex

- Videotex : serviço mais sofisticado que usa linhas telefônicas. Usuário loga em um sistema de tempo compartilhado controlado pelo provedor do serviço de videotex.
- Informação é enviada de acordo com a necessidade do usuário
- Informação pode ser maior que a enviada pelo teletext

9.5.4 - Teletext e Videotex



9.6 - Exemplos de camada de aplicação

9.6.1 - Redes públicas

- Transferência, acesso e manipulação de arquivos :
 - FTAM (OSI)
 - Tipo de arquivo básico: hierárquico
 - Orientado a conexões
 - Bloqueio compartilhado
 - Esquema de proteção geral é complexo
 - Dividido em 9 subconjuntos devido à complexidade, em que um é obrigatório e o resto é opcional.

- Correio Eletrônico :
 - MOTIS
 - Não usa endereçamento do tipo nome@dominio
 - Usa o mesmo tipo de nome usado pelo serviço de diretórios O/R (Originator/Recipient)
 - Para envios internacionais um campo com o nome do país é adicionado
 - Reliable Transfer Service (RTS), serviço confiável para evitar perda de transferência.
- Terminais Virtuais :
 - VTS
 - Suporta modo sincrono e assincrono
 - Conceptual communication area, estrutura de dados compartilhada
 - Armazenamento de dados conceitual: array de caracteres com atributos visuais
 - Objeto de Controle : funções que não lidam com texto
 - Objeto Dispositivo : um para cada i/o ligado ao terminal
 - Comunicação Conceitual: lida com tokens
 - Objeto que lida com os parâmetros da comunicação

- Serviço de Diretórios
 - Experimento
 - Padronização de nomes e atributos comuns
- Transferência de Jobs
 - JTM
 - Documento contém vários arquivos
 - Manipulação de documentos
 - Interação com agências
 - Agência de Iniciação: processo que submete o trabalho
 - Agência de Origem: sistema de arquivos que provêm os arquivos necessários
 - Agências Locais: sistema de arquivos que aceitam o armazenamento dos arquivos
 - Agências de Execução: máquina que rodam os jobs
 - JTM usa de primitivas para que as agências funcionam em conjunto

9.6.2 - ARPANET

- Desenvolvidos ‘bottom up’.
- Dificuldade dos protocolos evoluiu de acordo com a necessidade.
- Designers tiveram pouca experiência prática de modo a incluir serviços que eles pensaram que seriam úteis.

- Transferência de arquivos :
 - Transferir de uma máquina a outra, sem armazenamento virtual
 - Considera a diferença entre as máquinas, que é ignorado no OSI
 - Distingue entre , imagem, ASCII, EBCDIC e arquivos binários
 - Reconhece alguns tipos estruturados
 - Modos de Transferencia:
 - Modo de Fluxo: arquivos comuns
 - Modo Comprimido: arquivos baseados em registros
 - Modo de Bloco: arquivos paginados.
 - NETwork BLock Transfer (NETBLT) protocolo que funciona logo acima do IP, ignorando as camadas superiores

- Correio eletrônico :
 - Simple Mail Transfer Protocol (SMTP)
 - RFC 822 é o documento de definição
 - Inicialmente apenas mensagens contendo texto ASCII
 - Não faz distinção de envelope e mensagem
 - E-mail é um arquivo com headers
 - Endereçamento nome@dominio
 - EDU (educational institutions), COM (companies), GOV (government), MIL (military sites), ORG (nonprofit organizations) e dois ou três caracteres para abreviação do país
- Terminais Virtuais :
 - TELNET
 - Modo de Rolagem
 - Preocupação com manipulação de simplex data streams
 - Sem conceito de estrutura de dados idêntica em ambos os lados
 - Conversão de caracteres para o padrão ASCII

9.6.3 - MAP e TOP

- Subset da camada 7 OSI
- ACSE,FTAM e serviços de diretórios completamente suportados
- CCR e JTM não são
- VTS suportado totalmente no TOP e parcialmente no MAP
- TOP suporta X.400 e MAP suporta Manufacturing Message Standard (MMS), que é pretendido para comunicação máquina-máquina
- OSI não define bem a interface entre usuário, programa e camada de aplicação
- MAP e TOP desenvolvidos para que programas rodem em máquinas diferentes

9.6.4 - USENET

- E-mail e news
- Desenvolvido para ser compatível com o serviço de e-mail da ARPANET
- News funciona como um quadro de avisos internacional eletrônico, com vários grupos cada um com uma variedade de assuntos
- Usuários de um grupo podem ler todas as mensagens postadas além de postar
- News usa de várias características do sistema de e-mail