



Camada de Aplicação

Modelo OSI (Open System Interconnection)

Grupo: Felipe Belisário, Eliabe Vinicius,
Adiel Prado, Leonardo Ferreira e
Breno Caldeira

Introdução

01

**Servidores de
Arquivos**

02

Correio eletrônico

03

ÍNDICE

04

Terminais Virtuais

05

Outras aplicações

06

**Exemplos da camada
de aplicação**

01

Introdução



A camada de aplicação

- Última camada do Modelo OSI, a mais próxima do usuário.
- Programas usuário, também conhecidos como aplicativos, utilizam serviços desta camada para comunicações.
- Certas aplicações são tão comuns que foram desenvolvidos padrões para garantir que todas utilizem o mesmo protocolo.



Introdução

- Há várias demandas e funções comum entre diversas aplicações.
- Transferência de arquivos.
- Correio Eletrônico.
- Terminais Virtuais.



02

Servidores de Arquivos



Transferência, Acesso e Manuseamento de Arquivos

- Uma das principais atividades em redes
- Um servidor de arquivos pode ser caracterizado em:
- Estrutura de arquivo
- Atributos de arquivo
- Operações de arquivo

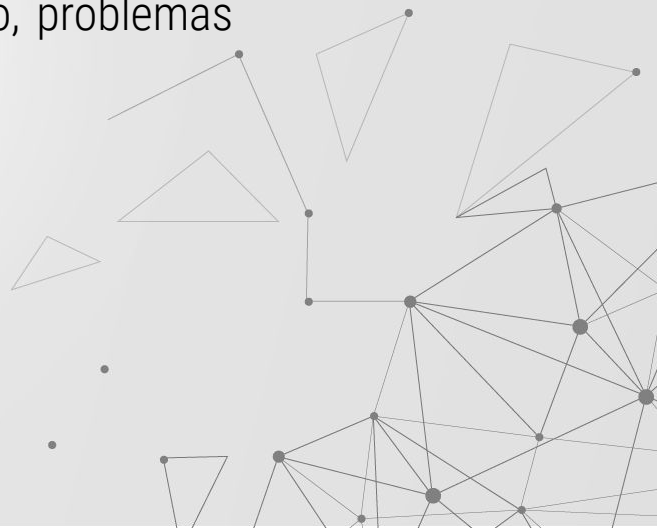


Attribute	Type	Set when file is created	User changeable	Maintained by server
File Name	String	X	X	
Allowed operations	Bit map	X		
Access control	List		X	
Account number	Integer	X	X	
Date and time of file creation	Time	X		
Date and time of last file modification	Time	X		X
Date and time of last file read	Time	X		X
Date and time of last attribute modification	Time	X		X
Owner	User id	X		
Identity of last modifier	User id	X		X
Identity of last reader	User id	X		X
Identity of last attribute modifier	User id	X		X
File availability	Boolean	X	X	
Contents type	Object id	X	X	
Encryption key	String	X		
Size	Integer	X		X
Maximum future size	Integer	X	X	
Legal qualifications	String	X	X	
Private use	String	X	X	



Controle de Concorrência

- Servidores de arquivos podem ter que lidar com múltiplos clientes.
- Se dois ou mais quiserem acessar o arquivo, problemas podem ocorrer
- Condição de corrida



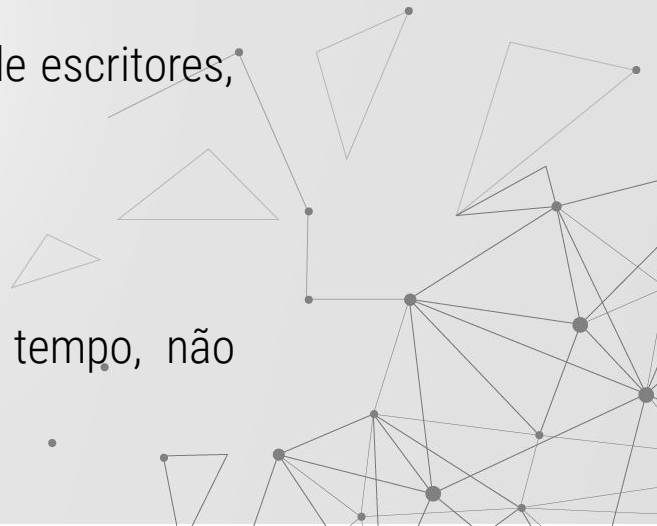
Soluções

- Trava exclusiva e compartilhada
- Na compartilhada, o acesso é liberado se o arquivo estiver livre ou com outras travas compartilhadas
- Usada geralmente para leitura
- Na exclusiva, apenas um usuário tem direito de acesso
- Ideal para escrita



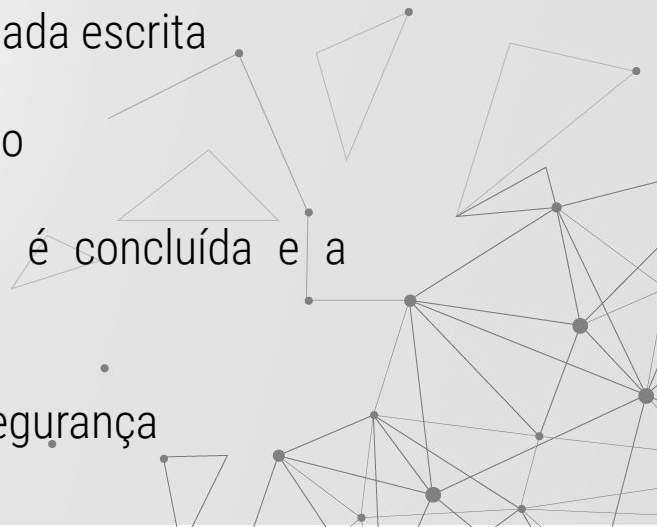
Capacidade de Acesso

- Múltiplos clientes podem ler do servidor
- Isso pode gerar problemas de congestionamento, além de rejeição de conexões
- O número de leitores pode ser maior que o de escritores, para aproveitar melhor a capacidade do canal
- É possível bloquear apenas parte do arquivo
- Dois usuários podem escrever ao mesmo tempo, não sendo na mesma região



Transações

- Alternativa para travas
- O cliente diz quando iniciar e finalizar uma transação
- Estado do arquivo é restaurado em caso de falha
- Implementadas através de cópias individuais para cada escrita
- Repõe o arquivo original quando sucede na operação
- Em caso de transações simultaneas, a primeira é concluída e a segunda abortada
- Poder computacional “desperdiçado” em troca de segurança



Master

```
Begin atomic action;  
Send Request 1;  
⋮  
Send Request n;  
Send "Prepare to commit" message;
```

```
if all slaves said "OK"  
  then send "Commit" message  
  else send "Rollback" message;  
Wait for acknowledgements
```

Slave

```
if action can be performed  
  then  
    begin  
      Lock data;  
      Store initial state on disk;  
      Store requests on disk;  
      Send "OK" message
```

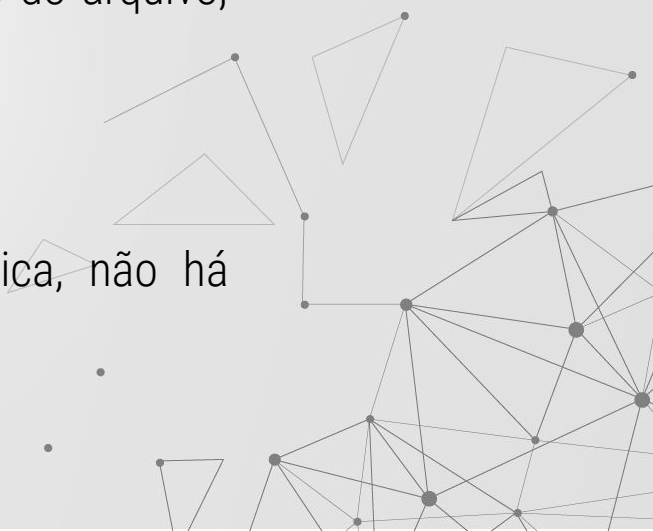
```
    end  
  else  
    Send "Failure"
```

```
if master said commit  
  then begin  
    Do work;  
    Unlock data;  
  end;  
Send "Acknowledgement" message
```

Fig. 9-2. Two phase commit.

Imutabilidade

- Vai na contramão da concorrência discutida previamente
- Quando um arquivo é criado, não pode mais ser alterado
- Para cada usuário é criada uma nova versão do arquivo, que não modifica a original
- O arquivo é repostado
- Criação de arquivo é uma operação atômica, não há necessidade de controle



Arquivos Duplicados

- Soluções propostas anteriormente serviam para sistemas de um servidor
- Redes complexas costumam ter múltiplos servidores para vários serviços
- Permite dividir o trabalho em vários servidores
- Permite continuar o acesso em caso de falha no servidor
- Aumenta a confiabilidade, múltiplos backups independentes



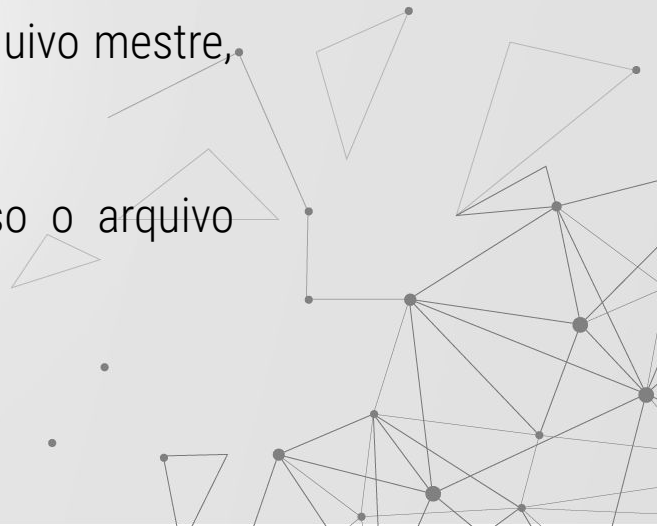
Métodos

- Poderíamos permitir ao usuário criar e operar em quantos arquivos ele quiser, e manusear as réplicas por conta própria
- Isso causaria uma experiência frustrante, principalmente entre leigos
- Há soluções que buscam solucionar esse problema



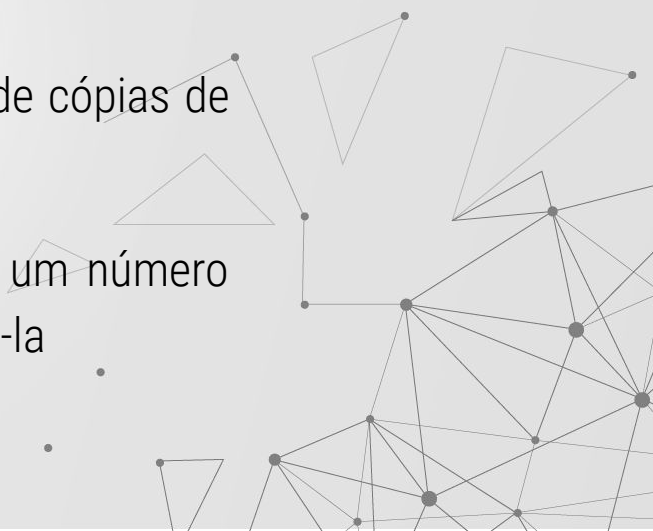
Réplica Primária

- Um arquivo é dito mestre
- A partir dele vem cópias, chamadas escravos
- Atualizações só podem ser realizadas no arquivo mestre, então são propagadas aos escravos
- Solução simples demais, serviço falha caso o arquivo mestre não esteja disponível



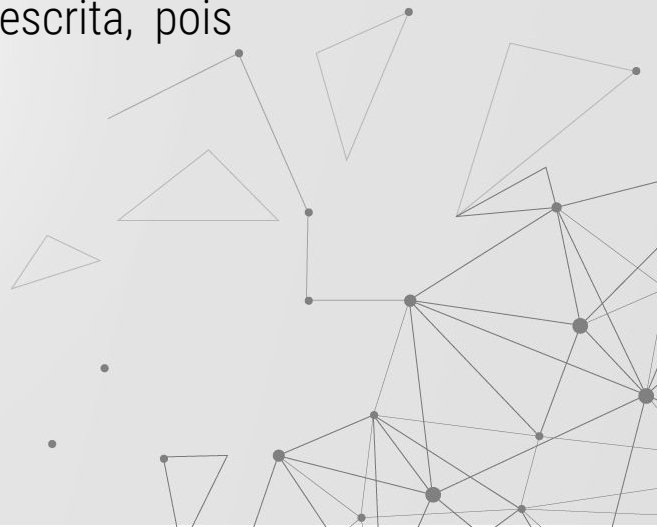
Votação

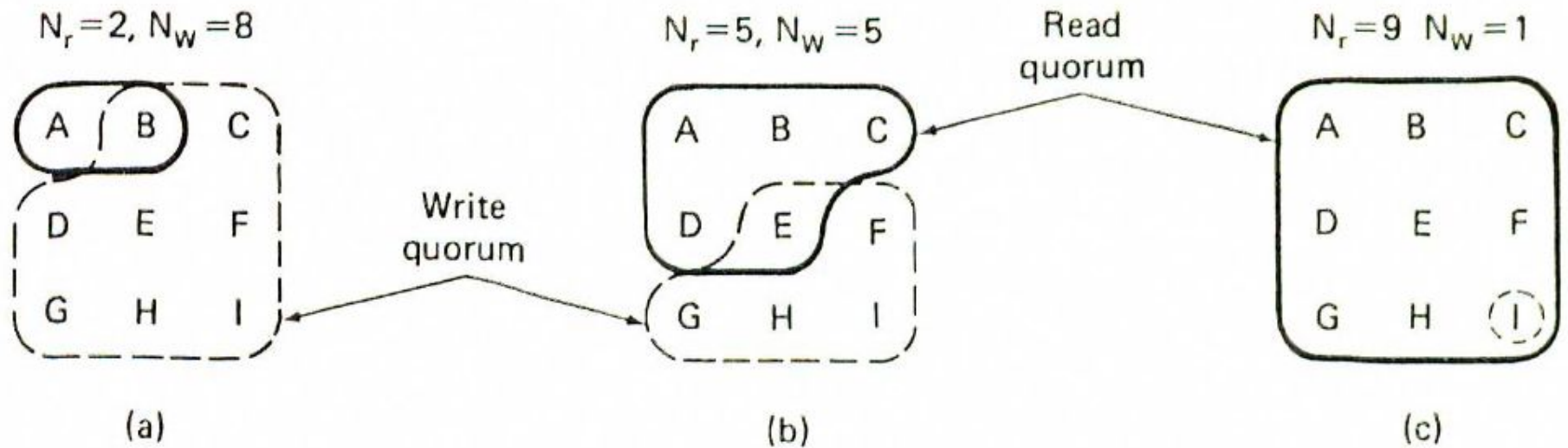
- Sistema mais robusto
- Para ler é necessário obter quórum de leitura N_r
- Para escrever, quórum de leitura N_w
- Condição, $N_r + N_w > N$, onde N é o número de cópias de um arquivo
- A operação de cópia só é realizada quando um número apropriado de servidores concordam e realizá-la



Exemplo

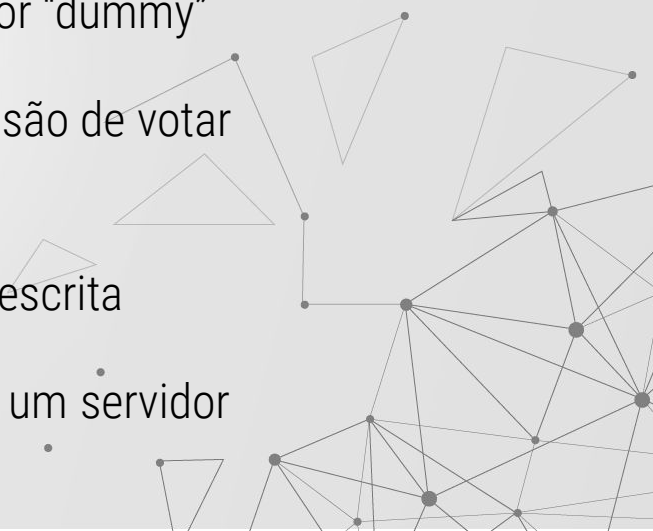
- Suponha $N = 9$, $N_r = 2$ e $N_w = 8$
- Um cliente adquiriu dois servidores, A e B, para leitura
- Outro cliente não conseguirá realizar uma escrita, pois não irá obter quórum de escrita
- Há nove deles ocupados, deverá esperar





Votação com Fantasmas

- Suposição: na maioria das aplicações, operações de leitura são mais frequentes
- N_w quase sempre se aproxima de N , enquanto N_r costuma ser um número pequeno
- Quando um servidor cai, é criado outro servidor “dummy”
- Este não possui armazenamento, nem permissão de votar em quórum de leitura
- Sua única função é votar no quórum de escrita
- Para a escrita ser bem sucedida, pelo menos um servidor deve ser real



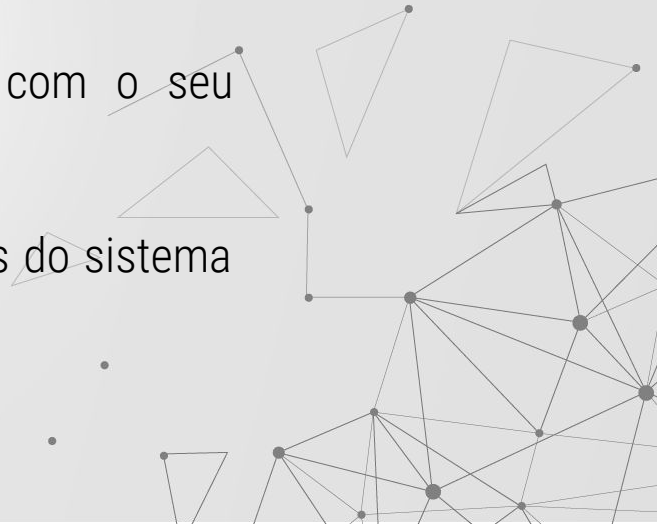
Problemas de Implementação

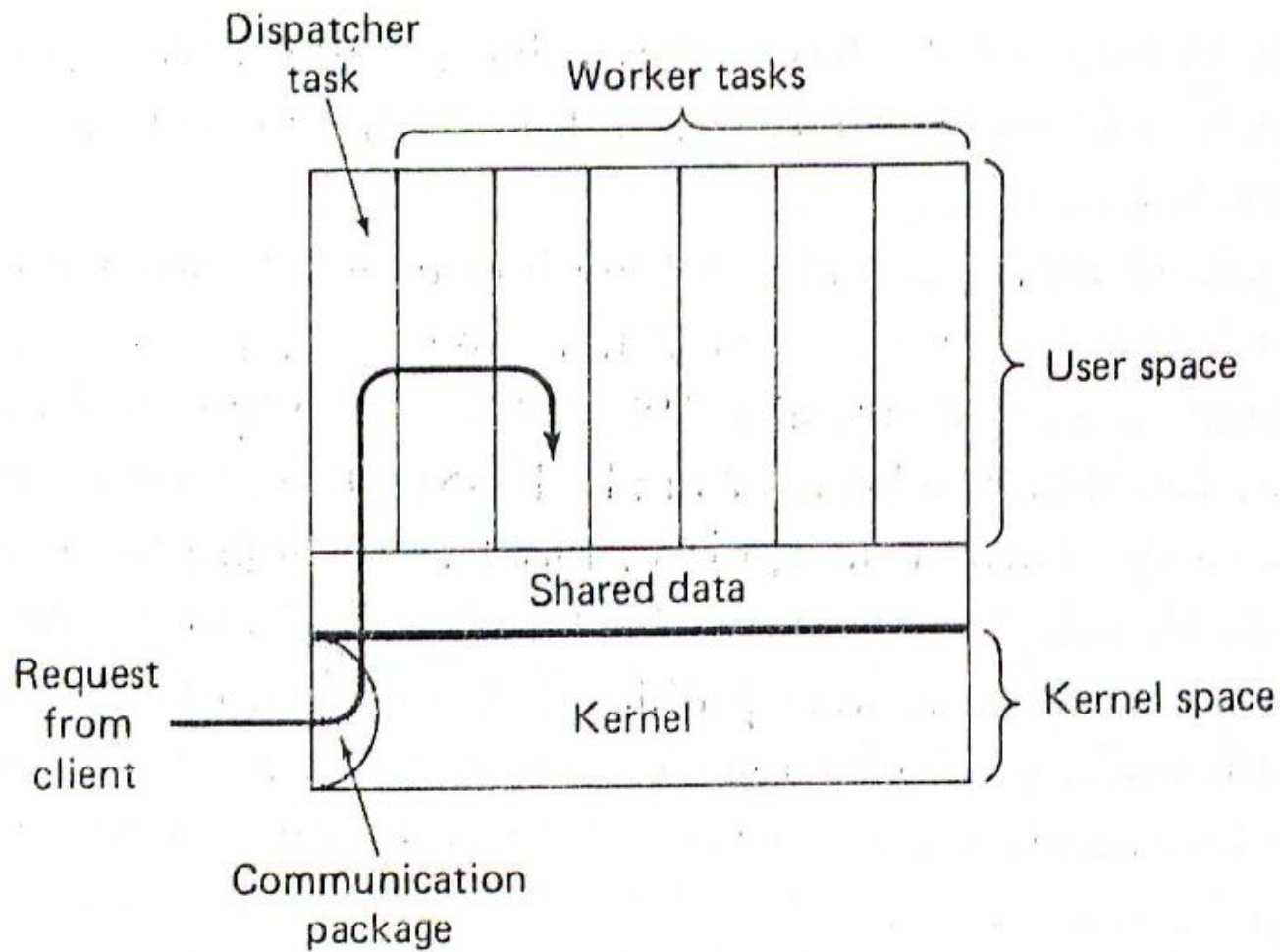
- Há várias maneiras de implementar um servidor de arquivos
- A mais simples consiste num loop principal, que recolhe, processa e responde à requisição, uma por uma
- Há várias desvantagens nessa forma, principalmente se uma requisição exigir múltiplos acessos ao disco



Despachante

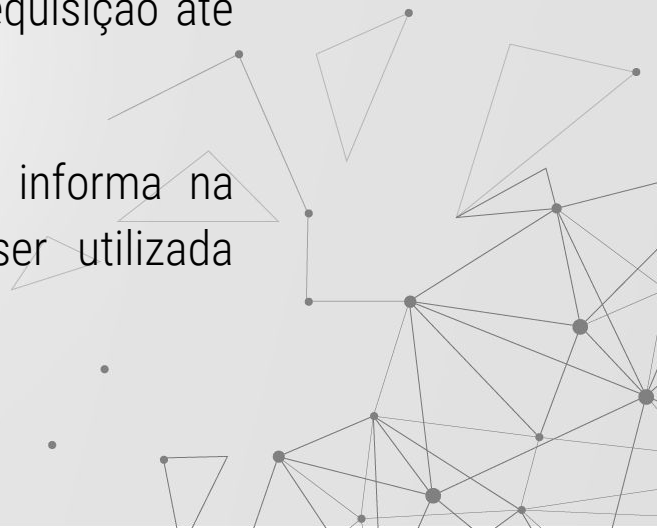
- Sistema mais sofisticado, que busca aumentar a performance
- O servidor é dividido em várias tarefas, que compartilha um espaço de endereçamento comum
- Cada tarefa possui sua própria “thread”, com o seu próprio “program counter” e sua pilha
- Tarefas dividem dados globais, como tabelas do sistema de arquivos e buffers





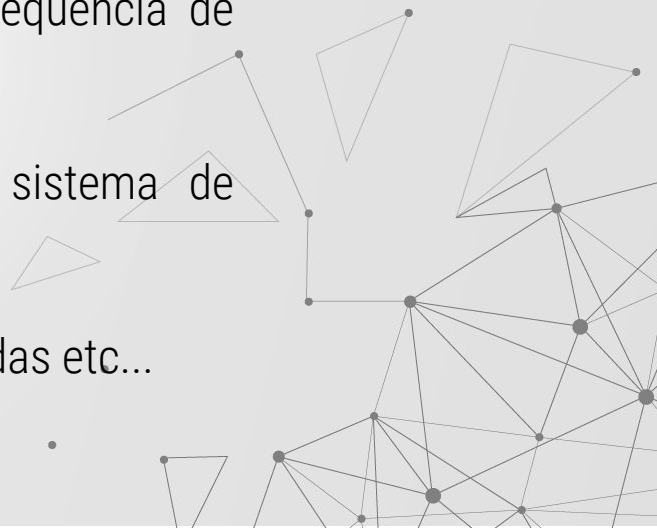
Funcionamento

- Quando um cliente requisita um serviço, o kernel aceita e passa para um despachante
- Este, que está esperando, analisa a mensagem através de um ponteiro à ele passado, e manuseia a requisição até completá-la
- Quando o serviço é concluído, a “thread” informa na tabela global que está disponível, para ser utilizada novamente



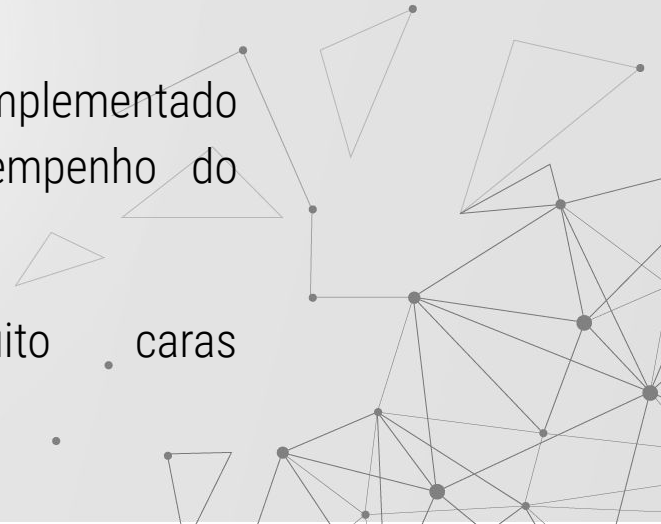
Organização

- A maioria dos sistemas de arquivos em servidores é orientado à blocos
- Um bloco é um conjunto de setores, e um arquivo, um conjunto de blocos, onde o setor é uma sequência de bytes
- Seu funcionamento interno varia com o sistema de arquivos implementado
- Pode ser usado mapa de bits, listas encadeadas etc...



Cache

- Para melhorar a performance, buffers de cache são comumente implementados
- Nele, os n blocos mais recentemente utilizados são mapeados para a memória principal
- Um cache suficientemente grande e bem implementado pode melhorar exponencialmente o desempenho do servidor
- Operações em discos são muito caras computacionalmente



Concorrência em Sistemas Distribuídos

- A necessidade de escrita em múltiplos arquivos que estão espalhados em vários servidores introduz vários problemas
- Há algumas abordagens para contornar isso



- A mais simples é para simplesmente dizer para o usuário ser “cuidadoso”
- Outra abordagem mais séria, diz para rastrear quais clientes possuem quais blocos e arquivos em seu cache
- Quando há uma modificação, o servidor envia uma mensagem para todos os clientes que possuem os mesmos blocos realizarem as devidas modificações
- Outra solução sugere armazenar apenas arquivos inteiros em cache, e fazer com que o servidor rastreie todos os arquivos em cache, para leitura e escrita
- Se um segundo usuário quiser abrir o mesmo arquivo, o primeiro deverá atualizar o mesmo na memória secundária



Orientação à conexão

- A implementação do sistema de arquivos varia se o serviço for ou não orientado à conexão
- Caso seja, o sistema de arquivos terá estados
- Para cada abertura, o servidor rastreia a posição do arquivo, para que o cliente apenas diga qual arquivo deseja ler e em quantos bytes
- Num sistema não orientado à conexão, os arquivos não possuem estados
- Cabe ao cliente ler o arquivo sequencialmente, e manter informações sobre sua posição atual



03

Correio eletrônico



➤ Os primeiros se consistiam simplesmente em protocolos de transferência de arquivos.

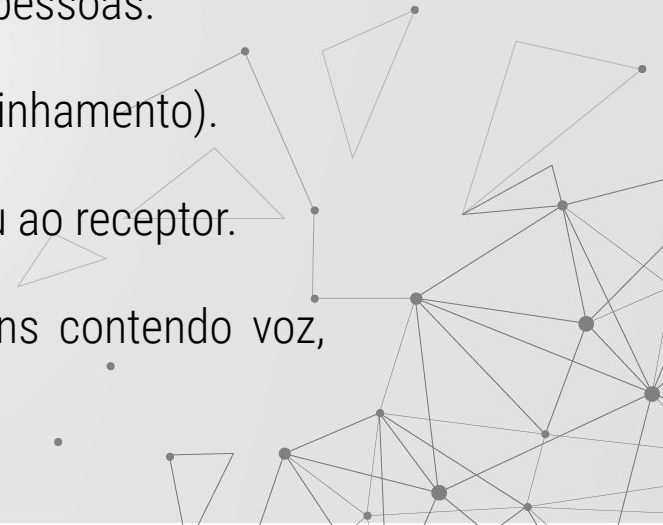
➤ Com o tempo as limitações dessa estrutura começaram a se tornarem óbvias:

1 - Inconveniente enviar mensagem para um grupo de pessoas.

2 - Mensagens sem estrutura interna (dificulta encaminhamento).

3 - Transmissor nunca saberá se a mensagem chegou ao receptor.

4 - Sem possibilidade de transmissão de mensagens contendo voz, desenhos e dentre outros.



- Novas propostas para sistemas de correio eletrônico mais ambiciosos.
- Em 1984, o CCITT (Comitê Consultivo Internacional de Telegrafia e Telefonia) elaborou a recomendação do X.400.
- X.400 foi utilizado mais tarde como base para o MOTIS do OSI.

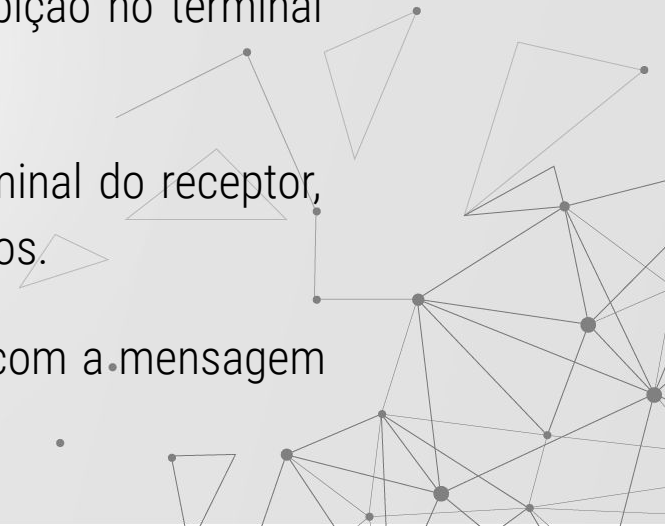


Arquitetura e Serviços do MOTIS e do X.400

- Tudo que for citado aqui sobre o MOTIS vale para o X.400 também.
- MOTIS se relaciona com todos os aspectos do sistema de correio eletrônico.
- A seguir será apresentada uma descrição sucinta dos 6 aspectos básicos de qualquer correio eletrônico.



- **Composição:** Processo de criação de mensagens e respostas.
- **Transferência:** Movimentação de mensagens do transmissor para o receptor.
- **Relatar:** Informar transmissor do que aconteceu com a mensagem enviada para o receptor (rejeitada, entregue ou perdida).
- **Conversão:** Tornar a mensagem adequada para exibição no terminal ou na impressora do receptor (incompatibilidade).
- **Formatação:** Formato da mensagem exibida no terminal do receptor, pode ser reformatada para cumprir aspectos desejados.
- **Disposição:** Se relaciona com o que o receptor faz com a mensagem após recebê-la.



- Uma ideia fundamental em todos os sistemas modernos de correio: distinção entre o envelope e seu conteúdo.
- O envelope possui parâmetros necessários para transporte e interpretação da mensagem.



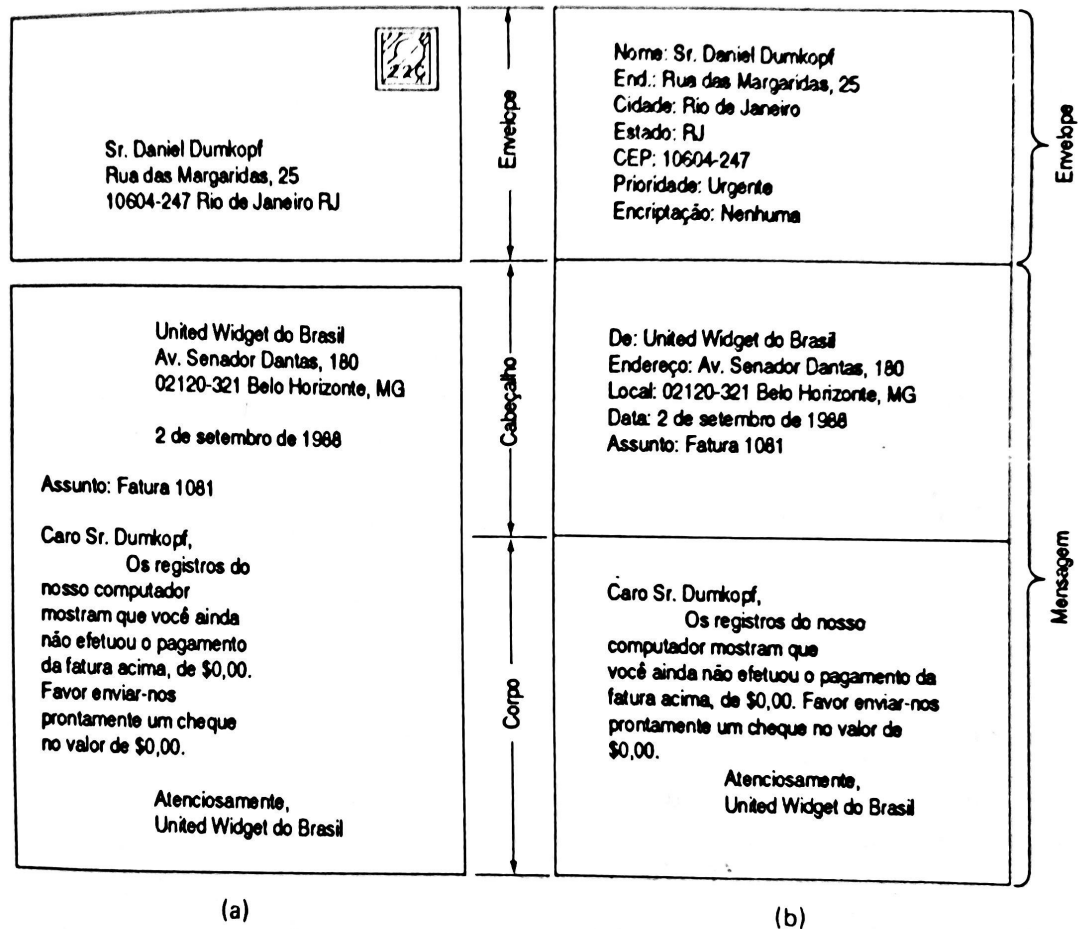


Figura 9-10 Envelopes e mensagens. (a) Correio postal. (b) Correio eletrônico

➤ Modelo utilizado pelo MOTIS

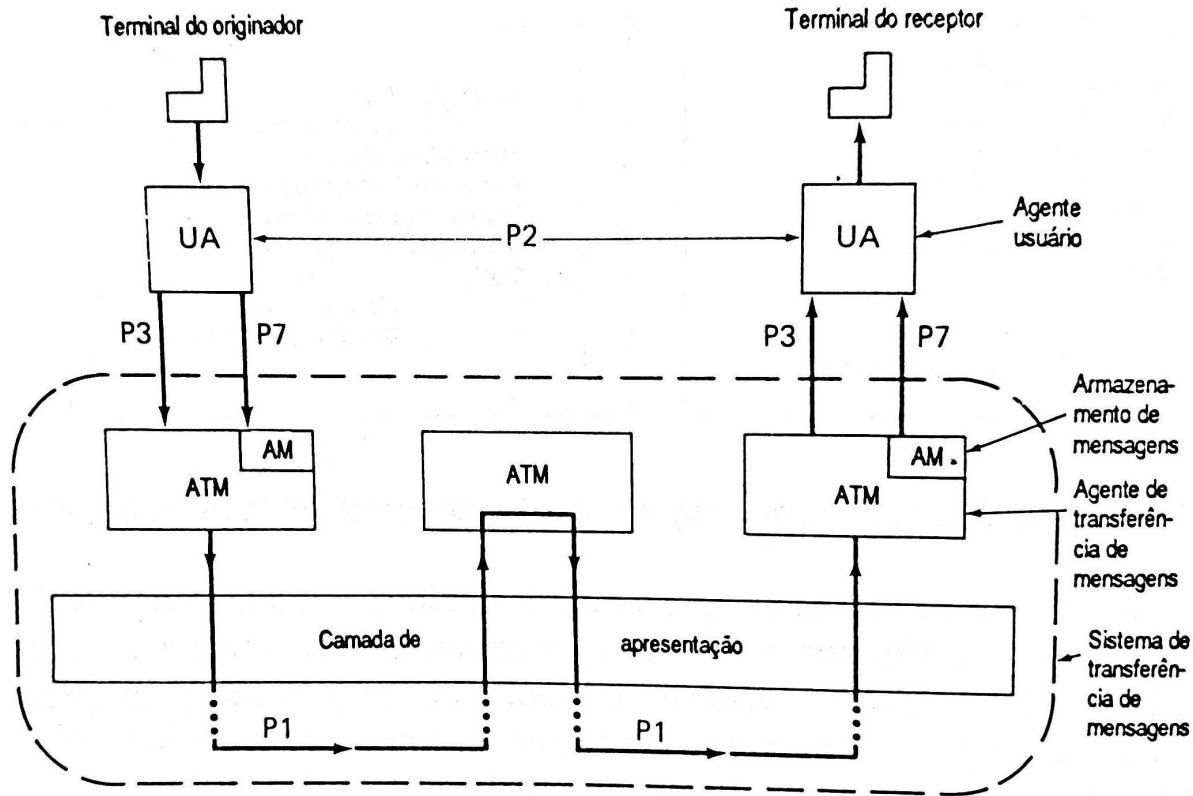


Figura 9-11 O modelo geral de um sistema de correio eletrônico

Agente Usuário

- É um programa que fornece a interface para o sistema de correio.
- Procura na caixa postal do usuário suas correspondências e espera comando para exibir elas.
- Cada linha de exibição contém diversos campos extraídos do envelope ou cabeçalho da mensagem correspondente.



#	Flags	Bytes	Emissor	Assunto
1	K	1030	Jonas	Bug no sistema de arquivos do MINIX
2	KF	2146	Ana L. Gomes	Solicita informações
3	KA	7136	RVR	Comentários em relação ao artigo sobre desempenho
4		3124	Erik	Aperfeiçoamento do modelo básico
5		610	Henrique	Encontro adiado até terça-feira
6		724	Jairo	Acho que a solução é 0-0
7		3240	Pedro Silva	Convite para vir conhecer os cangurus
8		425	Edith	Não se esqueça de preparar seu exame

Figura 9-12 *Um exemplo de exibição do conteúdo de uma caixa postal*

Comando	Parâmetro	Descrição
h	#	Exibe cabeçalho(s) na tela
c		Exibe cabeçalho atual na tela
t	#	Imprime mensagem(ns) na tela
s	endereço	Envia uma mensagem
f	#	Encaminha mensagem(ns)
a	#	Responde mensagem(ns)
d	#	Elimina mensagem(ns)
u	#	Recupera mensagem(ns) anteriormente eliminada(s)
m	#	Move mensagem(ns) para outra caixa postal
k	#	Mantém mensagem(ns) depois de sair
r	caixa postal	Lê uma nova caixa postal
n		Vai para a próxima mensagem e a exibe
b		Retorna à mensagem anterior e a exibe
g	#	Vai para uma mensagem específica mas não a exibe
e		Sai do sistema postal e atualiza a caixa postal

Figura 9-13 *Comandos típicos de tratamento da correspondência para o agente usuário*

- Vamos passar agora da interface do usuário para o protocolo utilizado entre agentes usuários.
- Esse protocolo é, em grande parte, definido pelos campos de cabeçalho incluídos em cada mensagem.

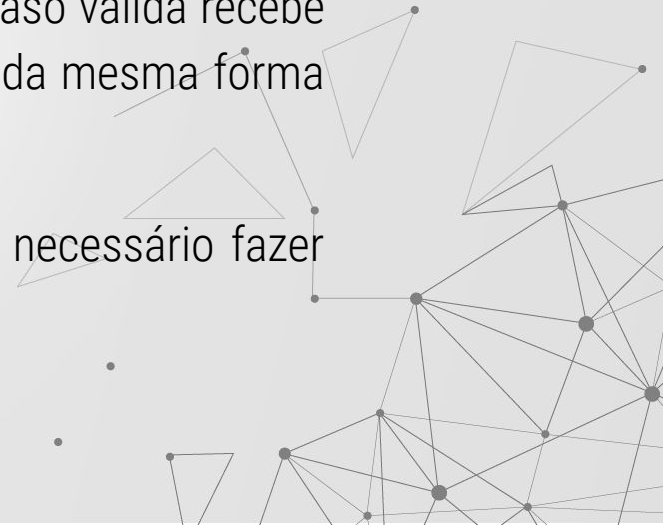


Campo	Descrição
Originador	Quem realmente enviou a mensagem
Usuários com autorização	Em benefício de quem ela foi enviada
Recipientes primários	A quem a mensagem é destinada
Recipientes de cópia	Quem recebe cópias em carbono
Recipientes de cópia oculta	Quem recebe secretamente cópias em carbono
Recipientes de resposta	Para quem deve ser enviada a resposta
Hora de resposta	Para quando se deseja a resposta
Id de mensagem	Identificador da mensagem
Em resposta a	Mensagem que gerou essa resposta
Mensagens obsoletas	Mensagens invalidadas por esta
Mensagens relacionadas	Outras mensagens relevantes para esta
Assunto	A que se refere a mensagem
Importância	Prioridade da mensagem
Sensibilidade	Pública, confidencial da companhia, etc.
Hora de vencimento	Hora em que a mensagem deixa de ser válida

Figura 9-14 *Alguns campos de cabeçalho para mensagens interpessoais*

Agente de transferência de mensagens

- Aceita correspondência dos agentes usuários e cuida para que a correspondência seja levada ao seu destino.
- Ao receber mensagem de agente usuário a sintaxe é verificada. Caso inválida a mesma é devolvida com uma explicação, caso válida recebe um identificador e uma estampa de hora e é tratada da mesma forma que uma mensagem que chega de outro ATM.
- Para entregar uma mensagem ao receptor pode ser necessário fazer uma conversão, se possível.



Campo	Descrição
Endereço do originador	Endereço postal do emissor
Endereço do recipiente	Endereço postal do recipiente
Recipiente alternativo permitido	Redirecionamento para outro recipiente permitido
Recipiente alternativo	Segunda opção de recipiente
Id da mensagem	Identificação da mensagem
Prioridade	Lento (econômico), normal, rápido (caro)
Pedido de relatório do originador	Que relatório o originador deseja
Pedido de relatório do ATM	Que relatório o ATM deseja
Entrega adiada	Não entregar antes desta hora
Entrega atrasada	Não entregar após esta hora
Retorno do conteúdo	O conteúdo deve ser devolvido se não for entregue
Tipo de informação	Texto, fac-símile, voz digitalizada, etc.
Conversão proibida	Nenhuma conversão é permitida
Conversão danosa proibida	Somente conversões perfeitas permitidas
Conversão explícita	Conversão sabidamente obrigatória
Identificação de encriptação	Índice na tabela de chaves de encriptação
Verificação da integridade do conteúdo	Total de verificação sobre o conteúdo
Assinatura do originador	Assinatura digital
Label de segurança da mensagem	Classificada, secreta, altamente secreta, etc.
Comprovação de entrega	Assinatura do recipiente

Figura 9-15 *Alguns campos do envelope*

04

Terminais virtuais



O que são os terminais virtuais?

➤ Os terminais se dividem em três grandes classes

➤ Scroll mode

➤ Page mode

➤ Form mode

➤ Para cada categoria, problemas diferentes estão presentes e abordagens diferentes são necessárias.



Scroll mode terminals

➤ São simples



Scroll mode terminals

- São simples
- Não possuem microprocessadores



Scroll mode terminals

- São simples
- Não possuem microprocessadores
- PAD (Packet Assembler/ Disassembler)



Page mode terminals

➤ Terminais CRT (Exibem 25 linhas de 80 caracteres cada)



Page mode terminals

- Terminais CRT (Exibem 25 linhas de 80 caracteres cada)
- Possuem os mesmos problemas do Scroll mode



Page mode terminals

- Terminais CRT (Exibem 25 linhas de 80 caracteres cada)
- Possuem os mesmos problemas do Scroll mode
- Tamanho de página, posição do cursor., etc



Page mode terminals

Code	Type	Description
bs	Boolean	Terminal can backspace with CTRL-H
hc	Boolean	Terminal prints hard copy
nc	Boolean	Terminal is a CRT but is not able to scroll
os	Boolean	Terminal can overstrike (multiple characters at same position)
ul	Boolean	Terminal can underline text
pt	Boolean	Terminal has hardware tabs

co	Integer	Number of columns in a line
li	Integer	Number of lines on the screen or on a page
dB	Integer	Number of millisecc of delay needed for backspace
dC	Integer	Number of millisecc of delay needed for carriage return
dN	Integer	Number of millisecc of delay needed for line feed
dT	Integer	Number of millisecc of delay needed for tab

cd	String	Clear screen from cursor to end of display
ce	String	Clear screen from cursor to end of line
cl	String	Clear entire screen
cm	String	Cursor motion
ct	String	Clear all tab stops
dc	String	Delete character under cursor
dl	String	Delete line containing cursor
ff	String	Go to top of next page (hardcopy terminals)
ic	String	Insert one character at cursor
is	String	String used to initialize terminal (and perhaps set tabs)
mb	String	Enter blinking mode
md	String	Enter bold (extra-bright) mode
us	String	Enter underscore mode
mh	String	Enter dim (half-bright) mode
so	String	Enter reverse video mode
me	String	Enter normal mode (no blinking, bold, dim, reverse video, etc.)
pc	String	Character to use for padding after tab, line feed, etc.
sf	String	Scroll screen forwards
sr	String	Scroll screen backwards

kl	String	Escape sequence sent by left arrow key
kr	String	Escape sequence sent by right arrow key
ku	String	Escape sequence sent by up arrow key
kd	String	Escape sequence sent by down arrow key
kh	String	Escape sequence sent by home key
kb	String	Escape sequence sent by backspace key

Fig. 9-17. A few of the more common entries in the termcap data base.



Form mode terminals

➤ É o mais sofisticado



Form mode terminals

- É o mais sofisticado
- Utiliza microprocessadores

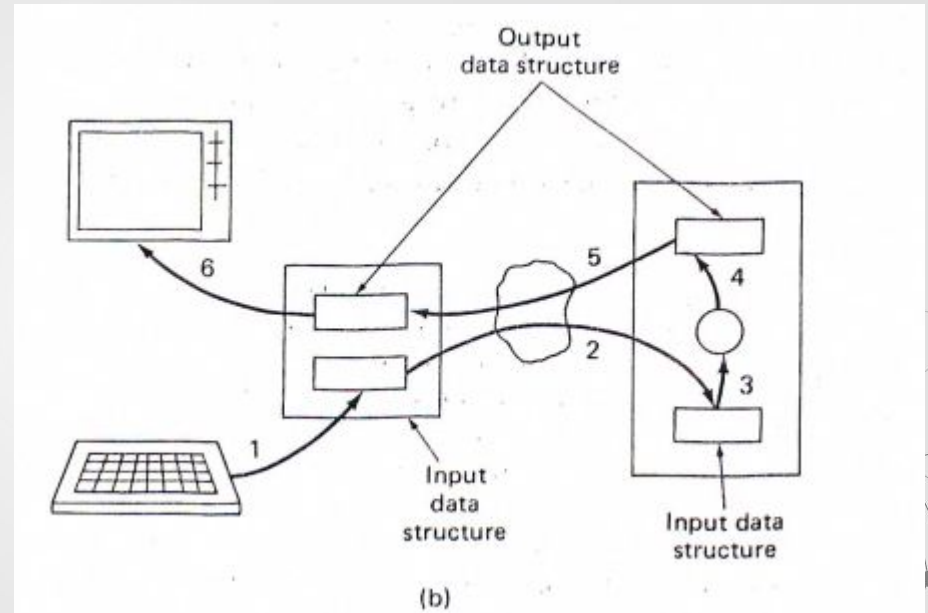
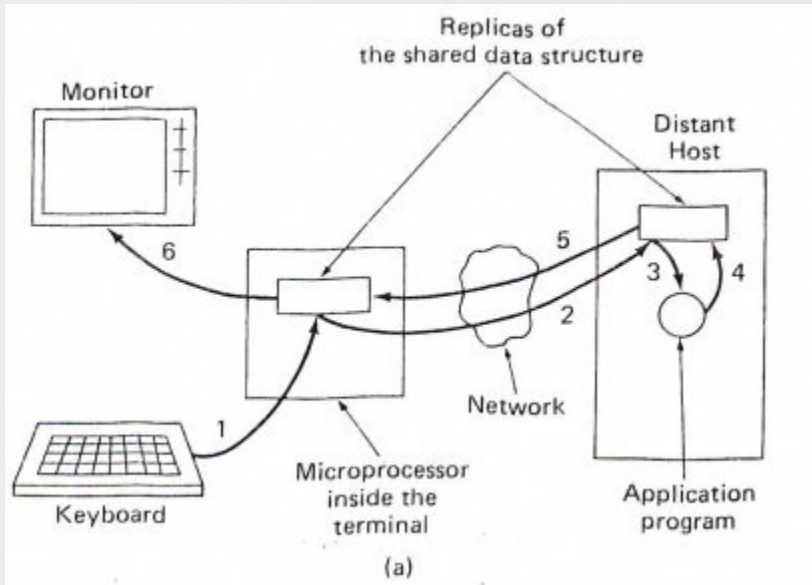


Form mode terminals

- É o mais sofisticado
- Utiliza microprocessadores
- Necessita de um modelo diferente de terminais virtuais baseado em estruturas de dados compartilhadas



Form mode terminals



Form mode terminals

Type	Description
Addressing	Move cursor to absolute position (x,y,z)
Addressing	Move cursor by relative amount ($\Delta x, \Delta y, \Delta z$)
Addressing	Move cursor to home position (1,1,1)
Text	Enter characters starting at current position
Attribute	Change attributes for the entire array
Attribute	Change attributes from cursor forward to (x,y,z)
Attribute	Change attributes from cursor backward to (x,y,z)
Attribute	Change attributes from (x1,y1,z1) to (x2,y2,z2)
Attribute	Change attribute mode on subsequent writes
Erase x	Erase this line from start to cursor
Erase x	Erase this line from cursor to end
Erase x	Erase this line from start to end
Erase y	Erase this page from start to cursor
Erase y	Erase this page from cursor to end
Erase y	Erase this page from start to end
Erase z	Erase from start of first page to cursor
Erase z	Erase from cursor to end of last page
Erase z	Erase from start of first page to end of last page

Fig. 9-19. Basic operations on the virtual terminal character array.

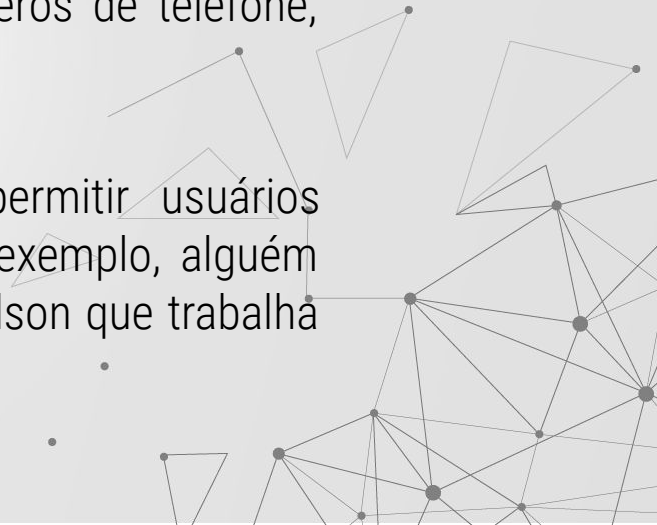
The background features a light gray surface with a network of thin, dark gray lines connecting various-sized dark gray circular nodes. These nodes are scattered across the frame, with a higher concentration on the right side, creating a complex web-like pattern. A single vertical line is positioned to the left of the main text.

05

Outras aplicações

Serviço de diretório

- Serviços endereçáveis sempre têm números endereçáveis grandes para não causar ambiguidade
- “No futuro isso (listas telefônicas e serviços de diretório) será substituído por diretórios online para procurar números de telefone, endereços de rede etc.”
- A ideia básica do serviço de diretório OSI é permitir usuários procurarem por nomes baseado em atributos. Por exemplo, alguém pode perguntar o número de uma pessoa chama hudson que trabalha no escritório da IBM em Los Angeles.



Serviço de diretório

➤ Criação de hierarquias

➤ C =EUA, ORG = SUN, DEPT=R&D, Nome= AL EINSTEIN

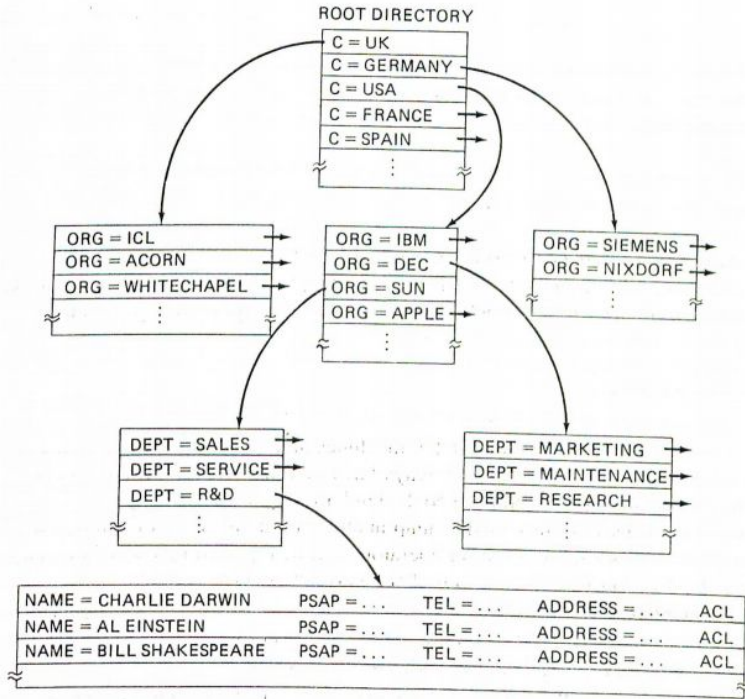
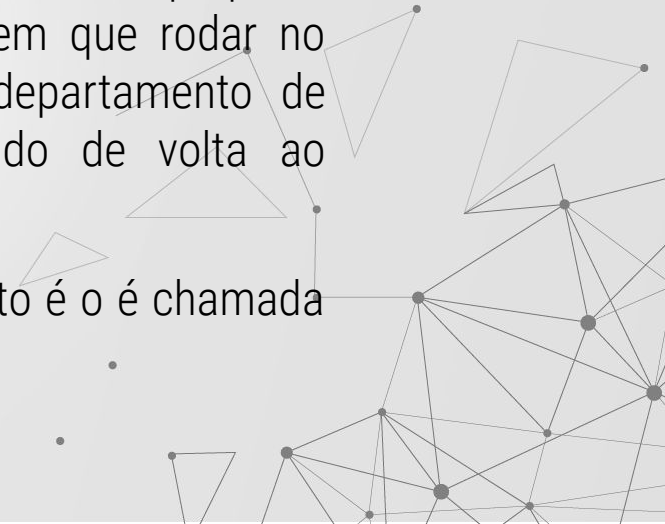


Fig. 9-21. An example directory hierarchy.

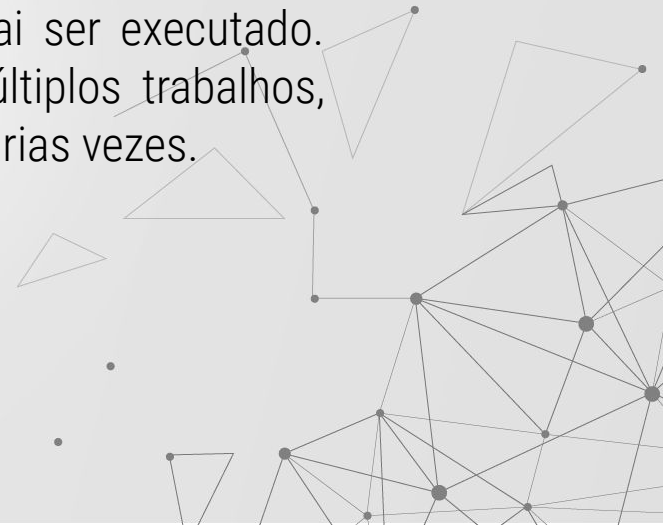
Transferência e gerenciamento de trabalhos

- “Em muitas grandes organizações, indivíduos têm computadores pessoais ou estações de trabalho em sua mesa, departamentos têm minicomputadores, e a organização como um todo tem um centro computacional com mainframes, supercomputadores e outros equipamentos caros. Ocorre frequentemente de um indivíduo preparar algum trabalho no seu computador pessoal que tem que rodar no mainframe usando arquivos localizados no seu departamento de minicomputadores com o resultado a ser enviado de volta ao computador pessoal”.
- A aplicação que gerencia esse tipo de trabalho remoto é o é chamada de JTM (job transfer and management).



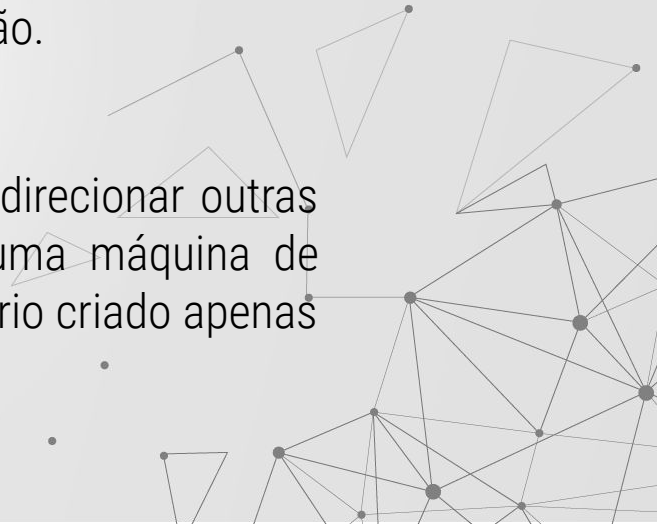
Transferência e gerenciamento de trabalhos

- Um cenário típico ocorre quando uma pessoa envia uma especificação de trabalho falando o que tem que ser feito, onde tem que ser feito, de onde as entradas vêm para onde os arquivos de saída vão. O JTM software tem que providenciar que o programa e todos seus arquivos de entrada acabem na máquina onde o trabalho vai ser executado. Especificações de trabalho podem consistir em múltiplos trabalhos, então o processo inteiro pode precisar ser repetido várias vezes.



Transferência e gerenciamento de trabalhos

- Uma maneira de enviar um pedido a cada máquina segurando um arquivo, e direcionar essa máquina a mandar o arquivo a máquina iniciadora (por exemplo usar o protocolo padrão de transferência de arquivo). Quando todos os arquivos e programas são coletados dessa maneira eles são enviados para a máquina de execução.
- Uma segunda maneira é a máquina de inicialização direcionar outras máquinas para enviar os arquivos diretamente a uma máquina de execução. Por exemplo, por os arquivos em um diretório criado apenas para esse propósito.



Transferência e gerenciamento de trabalhos

- Uma terceira maneira é ter uma especificação de trabalho movida a uma das máquinas que estão segurando arquivos, pegar os arquivos que precisa, e então mover para a próxima máquina. Essa migração continua até todos arquivos forem coletados, a cada momento todo o grupo continua se movendo nas máquinas de execução.



transferência e gerenciamento de trabalhos

- É importante reconhecer que o JTM sabe nada sobre o conteúdo dos arquivos, controle de linguagem de trabalhos, ou a natureza do processamento. Sua função é ver se os arquivos de entrada foram coletados, se os programas foram mandados para execução, e os arquivos de resultado são depositados onde eles deveriam ser. Ele sabe tão pouco sobre a natureza do processo assim como o FTAM sabe sobre os arquivos que transfere.



Armazenamento e transferência de imagens

- Como displays de alta resolução são comuns, terá pessoas interessadas em transmitir imagens como se fossem mensagens. Por essa razão muito trabalho tem sido feito em padronização de representação de imagens para elas poderem ser mandadas de diferentes lugares, aplicações e equipamentos.

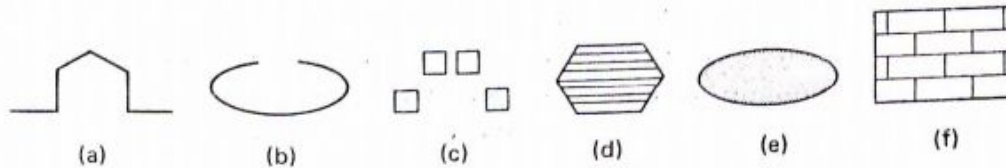


Fig. 9-22. (a) A polyline. (b) An elliptical arc. (c) A polymarker. (d) A filled polygon. (e) A filled ellipse. (f) A cell array.

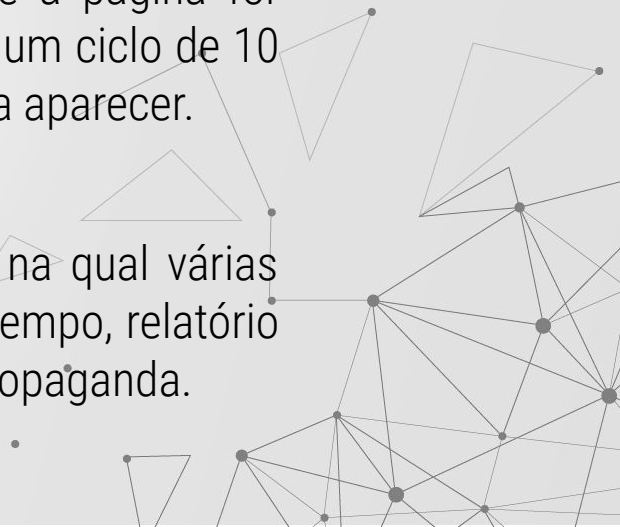
teletext e videotex

- Teletext é um serviço de via única na qual páginas de informação da base central de dados são transmitidas em um sinal regular de televisão
- Uma pequena parte do sinal da largura de banda é normalmente desperdiçada enquanto os feixes elétricos nos conjuntos de televisão são acionados na horizontal e vertical.



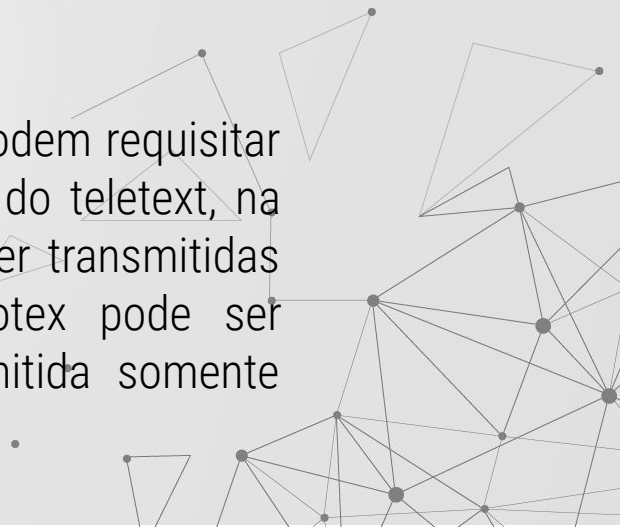
teletext e videotex

- Um sistema típico pode transmitir 100 páginas diferentes em um intervalo de 10 segundos, e então repete o padrão a cada 10 segundos
- Para usar o sistema, o usuário digita um número de página no painel junto ao decodificador de teletext. A próxima vez que a página for transmitida, é capturada do ar e colocada na tela. Com um ciclo de 10 seg, terá um delay médio de 5 segundos antes da página aparecer.
- Teletext é o mais adequado para informação simples na qual várias pessoas estão interessadas, por exemplo: previsão do tempo, relatório da bolsa, notícias, receitas, informação de produtos, e propaganda.



teletext e videotex

- Videotex é um serviço de 2 vias mais sofisticado usando linhas telefônicas.
- Em vários casos, o terminal é um teclado que gera um sinal de vídeo, para assim os conjuntos de cores padrões de televisão serem usados como display.
- Videotex é um serviço interativo no qual os usuários podem requisitar informações específicas. Diferente do base de dados do teletext, na qual é restrita a um número de páginas que podem ser transmitidas em alguns segundos, a base de dados do videotex pode ser arbitrariamente larga porque a informação é transmitida somente quando o usuário quer.



teletext e videotex

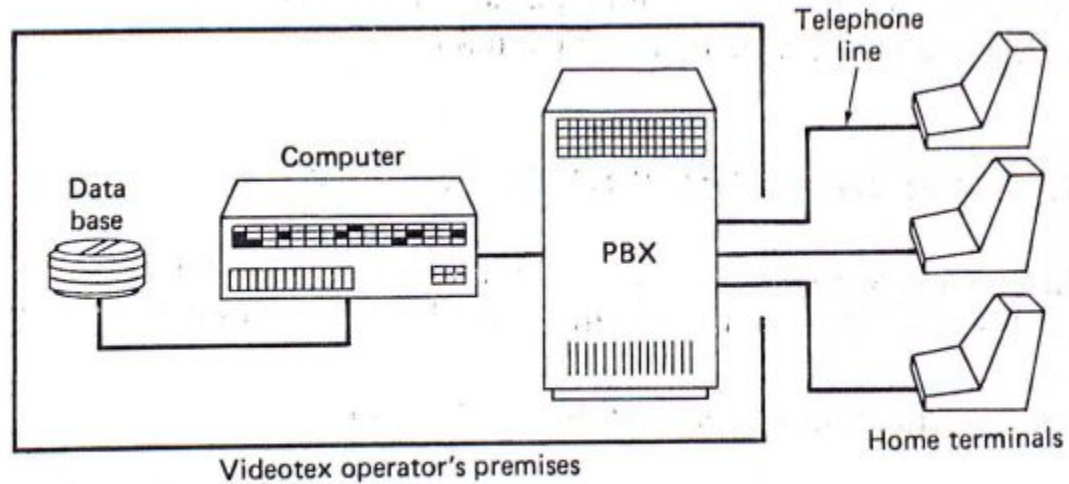
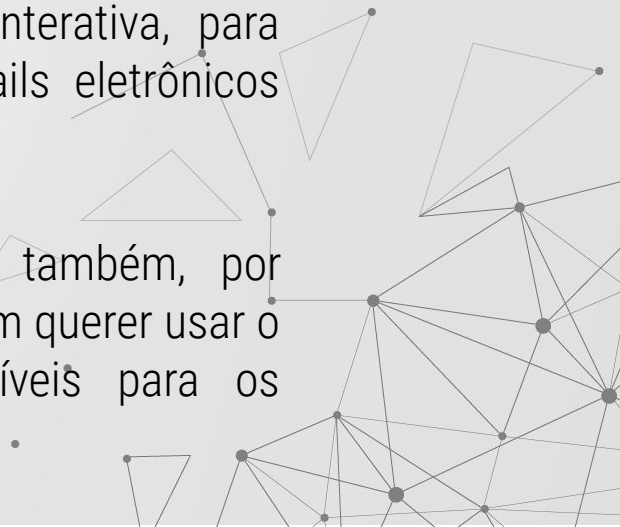


Fig. 9-24. Videotex is a two-way information dialup service.

teletext e videotex

- Videotex tem duas audiências em potencial: usuários de casa e de empresa.
- Usuários de casa podem estar interessados no serviço de informação, como linhas aéreas, ônibus, trem, tarefas, horário de filme etc. Eles podem também querer usar o sistema de forma interativa, para catalogar compras, fazer reservas etc. Mandar e-mails eletrônicos também é uma possibilidade.
- Corporações podem usar alguns desses serviços também, por exemplo reserva de hotel e avião. Além disso eles podem querer usar o sistema para deixar algumas informações disponíveis para os funcionários em nível nacional.



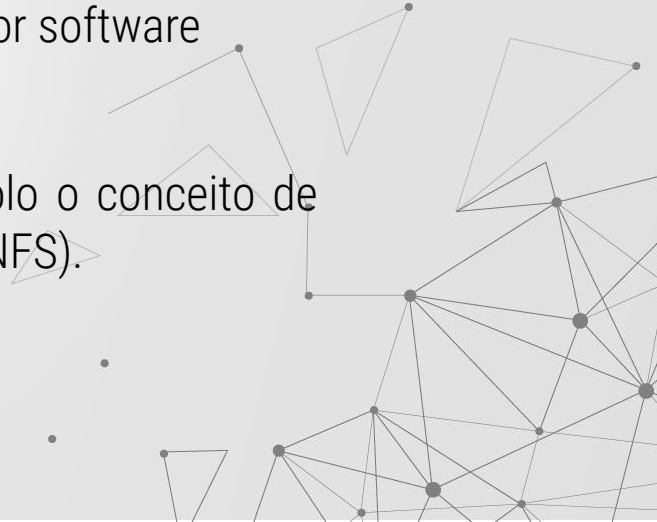


06

Exemplos da camada de aplicação

FTAM

- FTAM é um protocolo OSI que significa File Transfer, Access, and Management
- Baseado na ideia de um armazenamento de arquivos virtual que é mapeado para um armazenamento de arquivos real por software
- O objetivo do FTAM é colocar em um único protocolo o conceito de transferir arquivos (FTP) e acessá-los remotamente (NFS).



Primitivas do protocolo FTAM

FTAM Service primitive	C ^o	R ⁱ	Description
F-INITIALIZE	X	I	Establish a connection with the filestore
F-TERMINATE	X	A	Release a connection with the filestore
F-U-ABORT		all	User-initiated abort
F-P-ABORT		all	Provider-initiated abort
F-SELECT	X	A	Select a file for manipulation or transfer
F-DESELECT	X	S	Terminate a selection
F-CREATE	X	A	Create a file
F-DELETE	X	S	Destroy a file
F-READ-ATTRIB	X	S	Read file attributes
F-CHANGE-ATTRIB	X	S	Modify file attributes
F-OPEN	X	S	Open a file for reading or writing
F-CLOSE	X	O	Close an open file
F-BEGIN-GROUP	X	A	Mark start of an atomic action
F-END-GROUP	X	A	Mark end of an atomic action
F-RECOVER	X	A	Recreate open regime after a failure
F-LOCATE	X	O	Move file pointer to a specific FADU
F-ERASE	X	O	Destroy a FADU
F-READ		T	Read one or more FADUs
F-WRITE		T	Write one or more FADUs
F-DATA		T	Data arrival (F-DATA, indication only)
F-DATA-END		T	End of FADU (F-DATA-END, indication only)
F-TRANSFER-END	X	T	End of data transfer regime
F-CANCEL	X	T	Abruptly terminate data transfer regime
F-CHECK	X	T	Set a checkpoint
F-RESTART	X	T	Go back to a previous checkpoint

Fig. 9-26. The FTAM primitives.

Exemplo

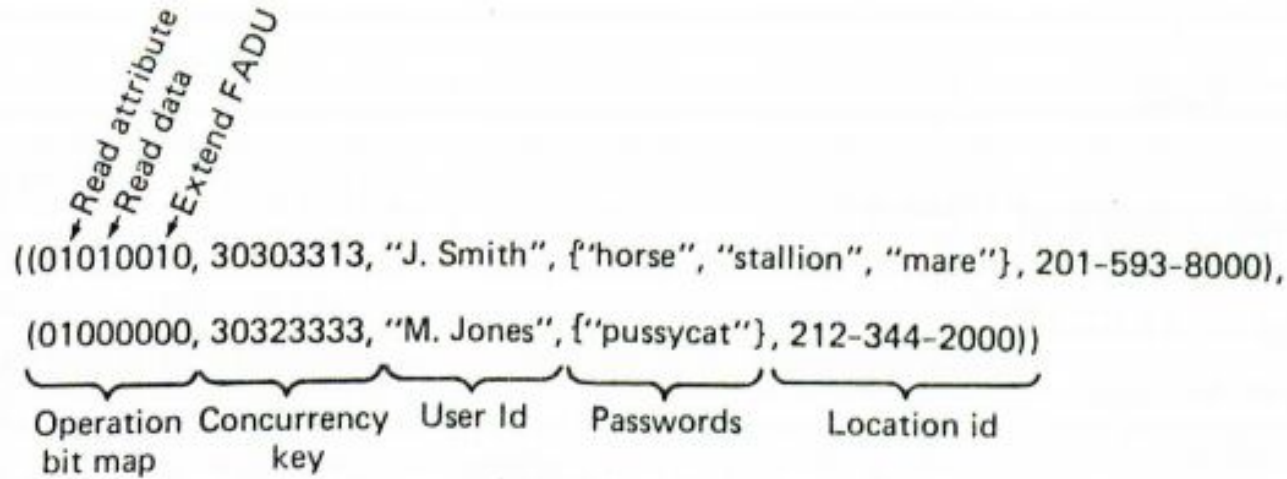


Fig. 9-27. An example access control attribute.

- A primeira lista dá permissão para ler os atributos e o dado do arquivo.
- A segunda lista permite ler os atributos, mas não permite leitura nem escrita de dados.

Email eletrônico

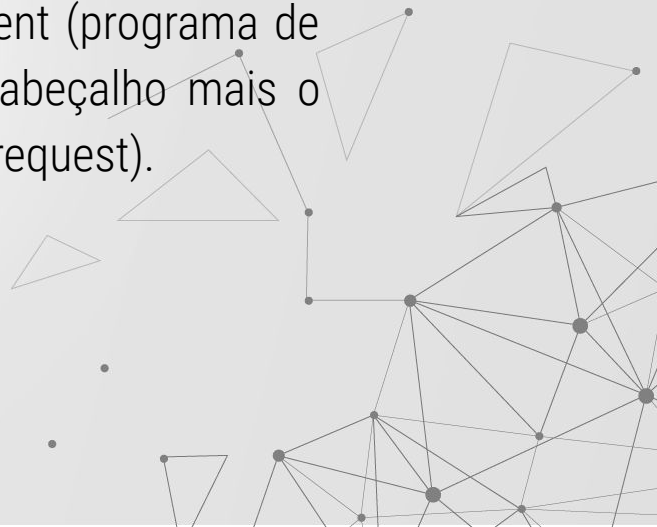
- O sistema de email eletrônico da OSI, o MOTIS, é baseado no X.400.
- Não é utilizado o padrão: nome@domínio como nos dias atuais. É usado o **O/R** (Origem/Recipiente), que é o mesmo tipo de nomes usado no serviço de diretório
- Por exemplo, um **O/R** seria do tipo:

ORG = Universidade Federal de Uberlândia, DEPT = FACOM, NAME = LUÍS FAINA



Email eletrônico

- Porém, esse **O/R** só será aceito aqui no Brasil, para tráfego internacional, o nome do país precisa ser especificado (mesmo que não exista outro lugar com o mesmo nome).
- O usuário constrói a mensagem usando um user agent (programa de email), que envia os campos mais a mensagem (cabeçalho mais o corpo do email), para uma primitiva de email (submit.request).





A camada de aplicação na ArpaNET

FTP - Transferência de arquivos

- O modelo de transferência de arquivos da Arpanet não é baseado na ideia de armazenamento virtual de arquivo como é feito com o FTAM.
- Este protocolo levou em conta o problema de transferir arquivos entre hosts com tamanhos diferentes da palavra, pois muitos destes hosts eram 36 bits, e outros eram 32 bits



FTP

➤ O Protocolo de Transferencia de Arquivos (FTP), distingue 4 tipos de arquivos: imagem, ascii, EBCDIC, e arquivos de bytes lógicos.

➤ Imagens: transferência de bit a bit

Ascii: arquivo padrão para transferência de arquivos, menos os main frames da IBM que usavam como EBCDIC

Arquivos de bytes lógicos: arquivos binários

➤ FTP suporta uma variedade de comandos para transferência de arquivos (enviar e receber), manipular diretórios, etc.



SMTP

- SMTP (Simple Mail Transfer Protocol) é o formato de email (definido no documento RFC 822) definido pela Arpanet.
- Foi criado com suporte apenas para caracteres ASCII.
- É usado o endereçamento do tipo nome@domínio. Os domínios nesse caso podem ser os TLDs (Top Level Domains).
- Criptografia: apenas o corpo do email é criptografado. Assunto, linhas de header estão em plain text.



Terminais Virtuais

- TELNET é o protocolo da Arpanet para terminais virtuais. Foi criado como modo de scroll ao invés de page ou form mode.
- O protocolo manipula duas streams de dados simplex, uma em cada direção. Não existe o conceito de ter uma estrutura de dados comum entre os dois hosts.



The background features a complex network of thin, light gray lines connecting various-sized dark gray dots. These dots are scattered across the slide, with a higher concentration on the left and right sides, creating a sense of interconnectedness and structure. The overall aesthetic is modern and minimalist.

THANKS

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.