

Quais os objetivos do amoeba?

Construir um sistema transparente ao usuário
Prover recursos para teste de aplicações distribuídas e paralelas.

Cite grupos de servidores existente no amoeba, explicando a sua funcionalidade

Bullet Server : servidores que mantêm arquivos (apenas os dados)
Directory Server : servidores que mantêm a localização dos arquivos.
Replication Server : servidores que gerenciam a replicação de arquivos
Run Server : Escalonadores dos processos.

Qual a relação entre gerência de memória e processos no amoeba?

Processos têm que residir completamente na memória. Eles não são paginados ou trocados.

Quais as funções do micro kernel do amoeba no que diz respeito a suporte de memória?

Da suporte à memória de baixo nível, por exemplo relação entre os endereços gerados pelo software e os endereços físicos endereçados pelo hardware (MMU)

Cite características comuns os kernel do Amoeba e Mach:

Gerenciar processos e threads
Prover gerenciamento de memória de baixo nível (fazer a troca de contexto)
Dar suporte a comunicação entre os processos
Gerenciar I/O de baixo nível

Qual a principal diferença entre o DCE e SO's como Amoeba, Mach e Chorus?

DCE opera em cima de um sistema operacional já existente (Unix, VMS, Windows,..)

DCE pode ser rodado em um ambiente heterogêneo?

Sim

DCE utiliza TCP/IP ou os protocolos do modelo OSI?

Pode usar os dois.

Qual modelo de comunicação que DCE usa?

Cliente/servidor. Processos usuários acessam serviços remotos disponibilizados por servidores.

Qual a diferença entre programas clientes e servidores?

Clientes são programas comuns com bibliotecas que fazem a comunicação, servidores são daemons.

Quais facilidades DCE disponibiliza para programação distribuída?

Threads e RPC.

Quais os serviços padrões do DCE?

Tempo, diretório, segurança e o sistema de arquivos distribuído em cima destes.

O que são células no DCE?

Agrupamentos de usuários, máquinas e outros recursos em um sistema DCE. Estes grupos geralmente espelham unidades organizacionais (departamentos, laboratórios,..)

Quais os fatores utilizados para formar células, explique-os brevemente.

Propósito: máquinas em uma célula devem ter um objetivo comum e seus usuários devem ter mais contato uns com os outros do que com usuários fora desta célula.

Segurança: usuários da célula tem acesso fácil aos recursos desta, mas difícil a recursos de fora.

Overhead: algumas funções DCE são otimizadas para operação intra-célula, localização geográfica afeta o desempenho e a confiança.

Administração: é mais fácil designar um administrador em um grupo menor e mais focado do que em um grupo que compreende dois departamentos distantes um do outro.

A tarefa de definir células é tecnológica ou administrativa?

Administrativa.

O que cada célula deve conter?

Um servidor de tempo, de diretório e de segurança.

As Threads DCE são baseadas em que padrão?

Posix p1003.4a

Se o sistema hospedeiro possui threads nativas, o DCE as utiliza?

Se configurado para isso, sim.

Quais os problemas relativos a utilizar threads neste sistema? p.528

Signals: threads podem não tratar signals (como exceções,...) e isso leva à morte do processo.

Reentrância de procedimentos: uma thread alocando recursos físicos é preemptada e a estrutura de dados que gerenciava este recurso fica inconsistente.

Sinalização de erros no Unix: errno é uma variável global, uma thread pode escrever nela e outra reescrever.

Qual a diferença entre escalonamento de processos e threads?

De threads é visível à aplicação.

Quais os algoritmos de escalonamento suportados por threads no DCE?

FIFO, Round-robin e o default (round-robin com prioridades)

Quais mecanismos de sincronização possíveis e quais tipos?

Mutex rápido: pedir um lock duas vezes leva a deadlock.

Mutex recursivo: permite que uma thread locks sobre mutexes já 'lockados' desde que sejam desfeitos.

Mutex não-recursivo: pedir um lock duas vezes retorna um erro.

Variáveis de condição: usadas junto com mutexes, usadas para coordenar recursos entre threads.

Pra que serve o função de Detach de uma thread?

Pra quando a thread pai não quer esperar o fim da thread filha. Quando esta termina, seu espaço de memória é recuperado automaticamente sem ter que esperar pelo join do pai.

Quais as chamadas para acordar variáveis de condição no DCE?

Signaling para acordar apenas uma ou broadcasting para acordar todas.

Que mecanismos de resistência as threads em DCE possuem?

Setcancel. Uma thread pode habilitar ou desabilitar a possibilidade de outras threads a matarem.

Quais as facilidades empregadas pelo RPC do DCE?

- localiza automaticamente o servidor e conecta

- faz o transporte da mensagem nas duas direções, fragmentando dados quando necessário

- converte automaticamente dados entre sistemas diferentes (por exemplo, little-endian / big-endian)

O que é um 'interface definition file'?

É uma listagem de todos procedimentos que o servidor permite que o cliente chame remotamente, contendo a assinatura destes procedimentos. É escrito com a Interface Definition Language (IDL).

O que resulta da compilação de um IDF?

- um header (por exemplo, interface.h) com as definições das chamadas, a ser usado tanto no código do cliente quanto do servidor

- o stub do cliente

- o stub do servidor

O que é o RPC daemon e pra que serve?

É um processo que roda em cada servidor que mantém uma base de dados contendo dados como <servidor, endereço de rede>, além de qual processo cada servidor disponibiliza. É utilizado para localizar servidores na hora de usar RPC.

Como um cliente acha um servidor - no modo mais simples?

O cliente pergunta pro servidor de diretório da sua célula onde tem um servidor. A partir daí, o cliente vai até o RPC daemon deste servidor e pergunta o endereço do servidor que realmente procura.

O que é e como funciona o DTS (Distributed Time Service)?

É um serviço do DCE que busca manter a sincronização de relógios entre máquinas.

Consiste em servidores de tempo que ficam verificando o horário um do outro constantemente. O DTS calcula a taxa de arrasto de cada clock e compensa isso através de cálculos.

Além disso, o DTS mantém os relógios em contato com a realidade, ou seja, com o horário real.

Como são registrados os horários no DTS? Qual o problema nisso?

Em intervalos, ao invés de dizer que são 9:52, o DTS diz que é entre 9:51 e 9:53. Ao pedir para comparar dois horários o DTS pode não saber qual é mais recente.

Quando é o início dos tempos de acordo com o DTS?

15 de outubro de 1582.

O que é o time clerk?

É um processo daemon que roda nos clientes e mantém o clock local sincronizado com os remotos. Ele também mantém um controle do crescimento linear da incerteza do clock local, quando este passa de um certo valor, o clock é resincronizado.

Como é feita a sincronização do cliente?

O time clerk contata todos os servidores de tempo na sua LAN, discarta os cujo tempo não interseccionam com os outros e calcula o ponto médio do horário retornado pela maioria. Este valor é atualizado gradualmente, mesmo que esteja adiantado ou atrasado, pois uma mudança brusca no tempo da máquina pode invalidar arquivos ou interferir com processos.

O que é o Cell Directory Service (CDS)?

É um serviço organizado em uma base de dados distribuída e replicada que provê boa performance e alta disponibilidade. Este serviço é o servidor de diretório que cada célula deve ter.

Quais os mecanismos disponíveis para localizar recursos?

Global Directory Service (GDS), é o padrão do DCE e é baseado em X500. E o Domain Name System (DNS).

Como interagem o CDS e o GDS/DNS?

Através de um terceiro componente do serviço de diretório do DCE, o Global Directory Agent (GDA), que é utilizado pelo CDS para interagir com o GDS/DNS. Quando o CDS precisa de um recurso, ele pede pro GDA que pede pro GDS/DNS.

Como são organizados os nomes no DCE?

Cada recurso possui um nome. O conjunto de todos os nomes forma o 'namespace' do DCE.

Cada nome pode ter até cinco partes:

- prefixo: indica se o nome é global ou local
- nome da célula: caso seja global
- nome:
- junção: equivalente a ponto de montagem
- nome da aplicação: nome do recurso

O que são 'principals' no DCE?

Processos usuário que precisam de comunicação segura. Por exemplo, usuários, CDS's e servidores de aplicação. Cada 'principal' tem um UUID (Unique User ID).

Como é o processo de autenticação no DCE?

Um usuário se loga pelo uso de um servidor de autenticação, por meio de Kerberos.

Como é o processo de autorização no DCE?

Cada recurso tem associado a ele, uma ACL (Access Control List), que indica quais usuários, grupos e organizações podem acessar tal recurso e de que maneira. A granularidade de recursos é variada.

Qual a ligação entre segurança e a estrutura de células?

Cada célula tem um serviço de segurança que contém o servidor de autenticação e mantém chaves, senhas e informações de segurança no registro.

Quais os requerimentos básicos de segurança levados em conta na criação do DCE?

- senhas não podem aparecer em texto comum na rede ou serem guardados em servidores comuns. Isto impossibilita autenticação pelo simples envio de senhas para um servidor de autenticação.
- senhas não podem ser guardadas em máquinas clientes por mais que alguns microssegundos
- autenticação vale nos dois sentidos, o servidor deve se convencer de que o usuário é quem diz que é e vice-versa
- o sistema deve ter firewalls 'de fábrica'

Quais os componentes do sistema de segurança do DCE?

- servidor de registro: gerencia a base de dados de segurança, o registro. Contem todas as informações

relativas a segurança para todos participantes da rede ('principals', grupos, organizações).

- servidor de autenticação: usado quando um usuário loga ou quando um servidor é iniciado. Também conhecido como 'ticket granting server', quando fornece tickets ao invés de autenticar usuários.

- servidor de privilégios: emite PACs (Privilege Attribute Certificates) para usuários autenticados. Estes PACs são mensagens codificadas com informações sobre a identidade de um 'principal' de uma maneira que servidores se convencem instantaneamente.

- 'login facility': programa que pede nomes e senhas à usuários quando estes estão fazendo login. Ele usa os servidores de autenticação e privilégios para garantir que o usuário seja logado e que tenha as permissões necessárias.

Ver Fig.10-26 (p.558)

O que é um ticket?

É uma estrutura de dados codificada emitida pelo servidor de autenticação para provar a um servidor que tal usuário é quem diz que é. Após o login, senhas não são mais utilizadas (para aumentar a segurança), apenas tickets.

A forma padrão é a seguinte:

ticket = S, {session-key, client, expiration-time, message-id} Ks

onde S é o servidor destino e K é a chave do servidor utilizada para codificar os dados entre chaves.

Desde o login do usuário até a primeira RPC autenticada, quais os passos necessários?

1. Cliente recebe um ticket do servidor de autenticação após fornecer sua senha.
2. Cliente usa o ticket anterior para conseguir um ticket para o servidor de privilégios.
3. Cliente pede ao servidor de privilégios o PAC inicial
4. Cliente pede ao 'ticket granting server' por um ticket usável pelo servidor de aplicação
5. Cliente estabelece uma chave com o servidor de aplicação

Ver Fig. 10-27

Como são gerenciadas as ACL?

Pelos gerenciadores de ACL, bibliotecas incorporadas em cada servidor. Estes servidores distinguem recursos em duas categorias: simples (arquivos e entradas de bases de dados) e containers (diretorios e tabelas de bases de dados). Também distingue entre usuários contidos em células e usuários externos.

Quais os direitos padrões disponíveis?

Read, write, execute, change-ACL, container-insert, container-delete e test.

Considerando-se que, no Sist. Op. Inferno, as aplicações utilizam recursos de maneira distribuída, como é possível que isso seja totalmente transparente para o usuário (desenvolvedor de aplicações)?

Através da utilização do protocolo de comunicação padrão denominado Styx (o que auxilia na questão da segurança), é implementada uma camada que funciona como mecanismo de comunicação do sistema, fornecendo acesso e apresentação de recursos, sejam eles locais ou remotos. Para os usuários, que não vêem o protocolo, mas apenas arquivos, existe uma transparência completa desses recursos (arquivos).

Sendo um sistema que pode ser hospedado em sistemas operacionais conhecidos (Windows, Unix, Plan 9...), existe alguma compatibilidade com protocolos de comunicações já conhecidos?

Na verdade, o que acontece é que, com o suporte do protocolo de comunicações Styx, o sistema trabalha de maneira independente e acima de vários protocolos de comunicação conhecidos (TCP/IP, ATM, PPP, etc).

No contexto da comunicação e segurança, o que o protocolo Styx provê para os usuários do sistema?

De certa forma, ele oferece uma visão hierárquica de recursos; informações de acesso como permissões, tamanhos e datas de arquivos (recursos) e semântica para leitura e escrita; além de fornecer mecanismos de criptografia de mensagens (com o conceito de canais de comunicação entre processos) e de autenticação de usuário (acesso aos recursos). O sistema dispõe de alguns algoritmos de criptografia presentes tais como SHA, MD4, MD5, Elgamal (assinaturas), RC4, DES, Diffie-Hellman (chave pública).

Quais são as principais semelhanças e diferenças com relação aos mecanismos de comunicação entre os sistemas amoeba, mach e chorus?

A principal semelhança é que todos eles tem suporte a RPC. As principais diferenças são que o paradigma principal de amoeba é comunicação endereçada a processo, enquanto chorus e mach tem comunicação endereçada a porta e que amoeba e chorus suportam comunicação a grupo enquanto mach só suporta comunicação ponto a ponto.

Como é a compatibilidade do Chorus com o Unix?

O Chorus utiliza um subsistema chamado MiX. A compatibilidade é em nível de código e em nível binário, ou seja, programas UNIX podem ser compilados e rodados no Chorus. O Mix também pode ler um sistema de arquivo Unix. Drivers de dispositivo também podem ser portados para o Chorus sem muito trabalho.

Quais os Objetivos do Chorus?

Emulação Unix de alta performance, uso em sistemas distribuídos, rodar aplicações em tempo real e integrar programação orientada a objetos.

O que são minimensagens e para que servem?

Mini mensagens são usadas em conjunto com as miniportas. São simplificações de mensagens e portas e são usadas somente entre processos do kernel para sinalizar a ocorrência de uma interrupção.

Explique de modo geral como funciona o escalonamento de processos no Chorus.

O Chorus escalona os processos usando prioridades e permite um processo ser iniciado em qualquer máquina. A sincronização de threads é feita por mutex e semáforos. Para permitir a execução de processos em tempo real, threads prioritárias só deixam a CPU quando terminarem ou quando ficarem bloqueadas. As outras threads utilizam round-robin.

Qual o problema apresentado pela gerência de memória do Chorus?

O Chorus dá suporte para memória distribuída compartilhada baseada em páginas. Quando uma thread referencia uma página que não está presente na máquina, esta página é trazida da máquina onde ela está. Esse mecanismo pode provocar thrashing.

Como é estruturado o Chorus?

O Chorus é estruturado por 4 níveis: microkernel, processos do kernel, processos do sistema, processos do usuário.

Quando foi criado, qual era o principal objetivo do sistema operacional Inferno?

Portabilidade. O Inferno tinha como propósito ser um sistema operacional distribuído que facilitasse o acesso a informações e recursos de uma maneira bastante portátil e simples.

Como se divide a arquitetura do Inferno?

Em três camadas: de aplicação, do kernel e de hardware.

Como funciona a camada de aplicação do Inferno?

É a camada de mais alto nível, sendo subdividida em:

- Aplicações Limbo: onde são escritos os programas na linguagem Limbo, que é semelhante a C e foi criada especialmente para o Inferno;
- Máquina Virtual Dis: Máquina Virtual com operações reduzidas que transforma as aplicações Limbo em código-objeto, onde posteriormente um compilador em tempo real faz a conversão das instruções Dis para o código de máquina do hardware utilizado.

O sistema Inferno tem grande aplicabilidade em sistemas embarcados, por quê?

Pois ele é bastante portátil e enxuto, podendo ser executado em ambientes com recursos de hardware mínimo.

Como funciona o controle de recursos do Inferno?

Todos recursos estão dispostos na forma de arquivos e organizados em um sistema hierárquico, desta forma, para utilizar um recurso basta manipular o arquivo referente a ele.