

Sistema Mach

Pergunta: Como surgiu o sistema Mach e quais são os principais objetivos do sistema?

Resposta: O sistema Mach começou na Universidade de Rochester, sob o nome de RIG - Rochester Intelligent Gateway - no ano de 1975. O principal objetivo do RIG era demonstrar que os sistemas operacionais podem ser construídos de uma maneira modular, sendo compostos por um conjunto de processos se comunicando por troca de mensagens. Esse sistema foi construído e apresentado, porém quando Richard Rashid saiu da Universidade de Rochester em 1979, ele continuou o mesmo projeto, agora com o nome de Accent. Este novo sistema possuía o acréscimo de operações transparentes sobre a rede, memória virtual de 32 bits, e outras pequenas diferenças.

Durante essa época, o sistema UNIX estava em grande crescimento, com isso, começaram a desenvolver compatibilidade com esse sistema e o 4.2BSD, a fim de garantir que todos os programas que foram desenvolvidos e estavam em desenvolvimento para o UNIX pudessem rodar no Mach. Em seguida a DARPA, a agência de desenvolvimento de projetos de pesquisa dos Estados Unidos, estava com interesse de desenvolver um sistema operacional que suportasse multiprocessamento e, com isso, surgiu o Sistema Mach. Este sistema é composto por um micro-kernel, que fornece os elementos básicos para o funcionamento de todo o sistema, por uma camada de usuário, na qual rodam os emuladores de sistemas operacionais e os controles dos sistemas de arquivos. Com essa estrutura, o Mach consegue ser um sistema altamente portátil, com um kernel extremamente rápido, até sendo utilizado em alguns computadores IBM de tempo real, e uma ótima plataforma para desenvolvimento de sistemas operacionais.

Pela história do sistema Mach conseguimos perceber alguns objetivos principais, e que se mantiveram até os dias de hoje. Os objetivos são os seguintes:

- Prover uma base para desenvolvimento de outros sistemas operacionais(exemplo: UNIX)
- Suportar um espaço de endereços largo e espaço
- Permitir acesso transparente aos serviços de rede
- Explorar paralelismo no nível de sistema e de usuário
- Manter o Sistema Mach portátil para um grande conjunto de máquinas.

A base para emular outros sistemas operacionais é fornecida através da arquitetura do micro-kernel, que provém os serviços essenciais para a camada de usuário, na qual ficam os controles do sistema de arquivos e os emuladores dos outros sistemas. Na figura seguinte está o esquemático do micro-kernel do Mach.

P: Cite quais são as principais abstrações pertencentes ao micro-kernel e explique.

R: As principais abstrações do micro-kernel do Sistema Mach são as seguintes:

- Processos
- Threads
- Objetos de Memória
- Portas
- Mensagens

Os processos são a base onde o ambiente de execução acontece, pois um processo possui um espaço de endereços, pode possuir um conjunto de threads, possui algumas portas, e uma ou mais pilhas. A alocação de recursos acontecem em torno dos processos, pois todos os recursos estão, de uma maneira ou de outra, relacionados com os processos. Um processo por si, não realiza nenhum processamento, com isso, ele possui um conjunto de zero ou mais threads, as quais realizam processamento. As threads possuem algumas estruturas próprias, como PC e registradores.

Da mesma maneira de que os processos possuem um conjunto de portas, as threads possuem uma porta, para realização de comunicação entre processos e com o kernel. A criação e a destruição das threads é realizada pelo kernel, mas a maior parte das tratativas com as threads são realizadas no próprio espaço de usuário.

A parte do controle de memória é realizada pelo kernel, com a abstração de objetos de memória, a qual pode ocupar uma ou mais páginas. Este controle é extremamente flexível e é baseado no sistema de páginas, o qual é completamente transparente ao nível de usuário. A parte de código que controla a parte que relaciona o sistema com a máquina alvo encontra-se no código do kernel, e a parte que controla o sistema lógico de arquivos, encontra-se no espaço de usuário, tornando-se assim, um sistema altamente portátil. O controle de memória virtual é muito semelhante aos outros controles que funcionam com o sistema de paginação, porém, ele permite que os endereços virtuais sejam espaçados, graças ao conceito de objeto de memória.

O conceito de portas e mensagens consistem na quase totalidade do sistema de comunicação do Mach, pois com esses elementos pode-se ter troca de mensagens, RPC, byte streams e outros. As portas possuem algumas características importantes, como: uma fila de processos que querem escrever/ler naquela porta, lista de processos que estão bloqueados, esperando para escrever/ler na porta, ponteiro para o seu respectivo objeto no kernel, número máximo de mensagens, capacidades da porta, entre outras.

As mensagens do Mach possuem uma estrutura muito bem definida e podem ser trocadas através de uma chamada de sistema, chamada pela função *mach_msg*. Através das mensagens, pode-se trocar dados, memórias, informações, bloquear threads.

Questões sobre o DCE:

a) Qual a maior diferença entre o DCE e sistemas distribuídos como Amoeba, Mach ou Chorus?

R: A proposta do DCE (Distributed Computing Environment) é fornecer um conjunto de ferramentas que criam um ambiente que servirá como plataforma para rodar aplicações distribuídas em sistemas operacionais. Os sistemas distribuídos tradicionais buscam implementar, desde o kernel, um novo SO, já o DCE constrói esse ambiente para computação distribuída no topo de sistemas operacionais já existentes (e.g. Unix, Windows).

b) Explique quais são e o que fazem os principais serviços disponíveis no DCE.

R: O DCE é composto por quatro serviços principais:

- **Distributed Time Service:** busca resolver o problema de sincronização entre os relógios fornecendo uma noção global de tempo. No DCE a informação de tempo é representada como um intervalo.
- **Directory Service:** mantém a relação dos nomes e localizações dos diversos recursos disponíveis no sistema. Antes de acessar um recurso o cliente pergunta ao Directory pela localização desse recurso. É dividido em dois serviços diferentes: Global Directory Service(GDS) e Cell Directory Service(CDS). O primeiro(GDS) é responsável por encontrar as diferentes células, o segundo(CDS) é responsável por encontrar os nodos dentro da própria célula.
- **Security Service:** Garante que o acesso aos recursos do sistema seja feito somente por clientes autorizados. Os clientes autenticados recebem PACs(Privilege Attribute Certificates) para que possam se identificar aos servidores (e vice-versa) sem precisar passar sua senha pela rede. A comunicação cliente-servidor é então feita por RPCs autenticadas.

- **Distributed File Service:** provê um sistema de arquivos onde o acesso a qualquer arquivo do sistema é feito de forma transparente, independente da sua localização. A concessão ao acesso dos arquivos é feito por um sistema de tokens.

Questões sobre I/O:

P: Qual o papel do Gerente de Fluxo (Streams Manager) no sistema Chorus? E como é seu funcionamento?

R: O gerente cuida das operações de I/O no sistema. Ele gerencia camadas de dispositivos como mouse, teclado, monitor. Quando o Gerente de Fluxo é inicializado, ele envia uma mensagem para o Gerente de Objeto (Object Manager) contendo a porta que ele está escutando e quais dispositivos de I/O ele está pronto para gerenciar. A partir disso, todas as chamadas de I/O são enviadas para o Gerente de Fluxo.

P: Faça uma breve descrição, considerando as operações de I/O, dos sistemas Amoeba, Mach, Chorus e DCE.

R:

O sistema Amoeba precisa de um driver para cada dispositivo de I/O que ele suportar. Todos os drivers que o sistema suporta já estão inclusos em seu kernel, então não é possível desenvolver drivers para suportar novos dispositivos de I/O. Ainda, toda a comunicação com os drivers de I/O é feita através do protocolo mensagens padrão. Finalmente, existe um servido específico para as operações de I/O.

Para o sistema Mach, o kernel disponibiliza uma série de serviços de I/O para o sistema. As threads de I/O são armazenadas em filas locais e a comunicação é toda feita através de portas.

Considerando o sistema Chorus, toda a operação de I/O é impossível de ser executada diretamente pelos processos de usuário, essas operações devem ser feitas por interrupção. A cada requisição de I/O é criada uma nova thread para tratar o evento. Além disso, o sistema Chorus suporta o mapeamento de segmentos do disco para a memória, possibilitando que o kernel leia e escreva diretamente na memória.

Finalmente, o sistema DCE realiza operações de I/O de forma assíncrona. Quando uma operação de I/O é terminada, uma notificação de evento é enviada para o sistema.

Questões sobre o Sis. Op. Inferno

a) Descreva os princípios da arquitetura do sistema operacional Inferno?

O sistema operacional Inferno, é um sistema operacional focado em serviços distribuídos, desenvolvido pela Bell Labs/Vita Nuova. Seus princípios de arquitetura são os seguintes:

Recursos como arquivos: Recursos locais e remotos como processos, armazenamento, rede, etc, são tratados como arquivos na árvore de diretórios do sistema, isso facilita a programação, já que teremos operações bem definidas como, abrir, ler, gravar e fechar. (Assim como no GNU/Linux)

Namespaces: Com o conceito de namespaces uma aplicação poderá ter um visão privada (instância local da aplicação em questão) dos recursos e serviços que ela precisa. Como os recursos são representados por arquivos, estes podem ser vistos como se estivessem em um único diretório raiz. A vantagem disso é que qualquer aplicação pode utilizar os recursos de modo transparente, uma vez que estes recursos estejam montados em um único namespace visível a esta aplicação, então diversos computadores podem compartilhar recursos entre suas aplicações como se fossem seus recursos locais.

Protocolo padrão para acesso de recursos:

O Inferno utiliza um protocolo implementado em kernel e nas aplicações para acesso e representação de recursos. Este protocolo chama-se **9P** (antigamente era chamado Styx). Esta abordagem prove uma maneira natural de construir sistemas distribuídos, utilizando esta “montagem” de diretórios. Só existe este padrão de comunicação e então há algumas soluções de segurança para comunicação das aplicações em cima deste protocolo:

- * Autenticação baseada em certificados;
- * Criptografia de mensagem.

Este protocolo roda sobre o TCP/IP podendo funcionar com outros protocolos de transporte também.

b) Como funciona o protocolo de comunicação de processo 9P(Styx) do Inferno?

O 9P(Styx) consiste de mensagens de 13 tipos para:

- * Iniciar comunicação (Anexar ao sistema de arquivos);
- * Navegar pelo sistema de arquivos;
- * Ler e escrever um arquivo;
- * Realizar consultas de estado de arquivo e mudanças.

Entretanto os programadores somente escrevem operações como open, read e write nos arquivos. Uma biblioteca ou o sistema operacional traduz as requisições na sequência necessária de bytes para transmitir ao canal de comunicação correto. O protocolo Styx é considerado como aproximadamente a camada de sessão do padrão ISO.

Uma chamada como:

```
open("/usr/rob/.profile", O_READ);
```

é traduzida em mensagens do protocolo 9P(Styx). Depois de estabelecida uma conexão como o servidor do arquivo, uma mensagem do tipo *attach* autentica o usuário e retorna um objeto chamado FID(File ID) que representa o diretório raiz do servidor.

Quando *open* é executado uma mensagem *clone* duplica a raiz retornando um novo objeto FID que pode navegar sem perder a conexão com a raiz.

Este FID clonado é movido para o arquivo *"/usr/rob/.profile"* por uma sequência de mensagens do tipo *walk* que “caminham” diretório por diretório.

Quando chegamos ao arquivo uma mensagem *open* checa as permissões para ler o arquivo e permite ou não as chamadas de leitura e escrita posteriores, uma vez terminada a leitura do arquivo uma mensagem *close* liberará o FID.

Questões sobre o Chorus

a) Explique brevemente cada uma das camadas que estruturam o SO Chorus.

R:

O sistema é estruturado através de 4 camadas. A camada mais inferior é o microkernel que provê o mínimo gerenciamento de nomes, processos, threads, memória e comunicação. Esses serviços podem ser acessados através de chamadas ao microkernel. A seguir pode ser visto uma figura que descreve todo o sistema de camadas:

Como pode ser visto, logo acima do microkernel existe a camada do kernel. Nessa camada processos podem ser carregados e removidos conforme a execução do sistema. Processos na camada de kernel em conjunto com o microkernel são os únicos que podem executar I/O e outras instruções protegidas.

A próxima camada contém os processos de sistema. Essa camada em conjunto com a próxima rodam em modo de usuário. Ou seja perdem o acesso privilegiado ao hardware(perdem o acessos a instruções protegidas e I/O). Essa camada ainda pode fazer chamadas diretas ao microkernel.

Acima vem a camada de processos de usuário. Essa camada não pode fazer chamadas diretamente ao microkernel, nem ao kernel, somente chamadas de sistema(a camada inferior) são suportadas.

b) Descreva brevemente o processo de gerenciamento de memória utilizado pelo SO Chorus. Não esqueça de explicar como é garantido o suporte a memória distribuída compartilhada:

O gerenciamento de memória é feito através de segmentos e regiões.

Uma região é uma série contínua de endereços virtuais. Regiões basicamente são partes da memória virtual que são "alocadas" para um certo processo e podem ser usadas pelo mesmo, ou seja, uma região é um dos recursos disponíveis ou uma característica de cada processo.

Um segmento é uma coleção contínua de bytes nomeados e protegidos por um identificador de recurso(capability). Segmentos podem ser lidos e escritos usando chamadas de sistema que disponibilizam os recursos para o segmento. Arquivos e swap são os tipos de segmento mais comuns. Entretanto, uma possibilidade é mapear segmentos em regiões. Segmentos mapeados são usualmente paginados por demanda(a não ser que essa função esteja desativada, por exemplo, para programas de tempo real), ou seja, ao ocorrer um "page fault" a página de segmento correspondente ao endereço referenciado é trazida para memória e a instrução que gerou o "page fault" é reiniciada.

O mapeamento de memória pode ser gerenciado por paginadores externos, que são chamados de "mappers". Além disso, o Chorus suporta memória compartilhada distribuída. Ele usa um algoritmo dinâmico descentralizado, o que quer dizer que diferentes gerenciadores mantêm o controle de diferentes páginas, e o gerenciador para uma página muda conforme a página se movimenta dentro do sistema.

Questões sobre sistemas de arquivos

O que é um Stateless File System? Diga uma vantagem que ele possui.

R: É um sistema de arquivos no qual o sistema não mantém na memória que um arquivo está aberto por um programa. Ele tem a vantagem de ser tolerante a falhas, pois não corre o risco de perder essas informações por um desligamento forçado.

Valeria a pena mandar cópias para todos os servidores de um arquivo pequeno, de aprox. 2KB, criado por um processo de usuário?

R: Na maior parte dos casos não. Existem alguns casos em que se pode prever com uma boa precisão que um arquivo é necessário somente na máquina que o criou; como no caso de pequenos arquivos temporários.

Questões sobre Comunicação

1 – No AMOEBA, como é feita a comunicação dentro de grupos? E entre grupos?

R - Dentro do grupo a informação é trocada por broadcast e memória compartilhada, já entre grupos, há informação é trocada por chamadas remotas de função (RPC)

2 – Como é feita a comunicação no sistema Chorus?

R – O mecanismo de comunicação entre processos (IPC) de Chorus permite que threads comuniquem-se através de troca de mensagens assíncrona ou usando remote procedure calls.

Questões sobre comparações gerais sobre os sistemas:

a) Descreva brevemente os Sistemas Operacionais Amoeba, Mach, Chorus, DCE e Inferno.

R:

- Amoeba é um SO transparente, que apesar da utilização de diversos computadores independentes, aparece como um tradicional sistema time-sharing para o usuário.

Devido à esta abstração, o usuário não sabe onde seus processos estão sendo executados.

Foi desenvolvido de tal maneira que não há um computador 'principal', o usuário loga-se ao sistema como um todo, não a apenas uma máquina.

- Mach é um SO baseado em microkernel. Seu propósito é servir de base para a construção de novos SOs e emular SOs já existentes.

Também proporciona flexibilidade para estender UNIX para ambientes com multi-processadores e sistemas distribuídos.

- Chorus é um SO baseado em microkernel. Fornece compatibilidade binária com System V UNIX, suporta aplicações de tempo-real e programação OO.

Consiste em 3 camadas conceituais: camada do kernel, o subsistema e os processos de usuários.

- O sistema operacional DCE fornece uma camada no topo do sistema operacional nativo, que esconde as diferenças dentre as máquinas, também fornece serviços comuns e facilidades que unificam o conjunto de máquinas em um único sistema que é transparente em alguns aspectos.

- O sistema operacional Inferno é baseado em três princípios de arquitetura: Recursos tratados como arquivos, Namespaces e Protocolo padrão para acesso de recursos.

b) Qual a principal diferença entre os sistemas Amoeba, Mach, Chorus e o sistema DCE?

R: Amoeba, Mach e Chorus são baseados em microkernel e fornecem uma base para outros sistemas rodarem em cima, enquanto o sistema DCE coloca uma camada em cima do SO nativo da máquina.

Questões sobre o Kernel

Cite uma desvantagem do Kernel implementado no sistema Chorus

R: Por ter partes implementadas em assembler, e por uma parte do controle de memória virtual precisar saber como fazer acesso aos registradores MMU, sua portabilidade é muito limitada, sendo necessárias modificações para ele poder ser portado a outras máquinas.

No sistema AMOEBA, como funciona a comunicação entre processos?

R: A comunicação pode ser ponto a ponto, baseado no modelo cliente servidor, um processo envia uma mensagem para o servidor e se bloqueia até que ele receba a resposta. E a comunicação também pode se dar em grupo, neste caso uma mensagem pode ser enviada para um grupo de destinatários, tendo suporte do sistema através de protocolos de software para se ter confiabilidade e tolerância a falhas.

Questões sobre Memória

1) Qual a diferença no gerenciamento de memória entre os sistemas Amoeba, Mach e Chorus?

- O modelo de memória do Amoeba é baseado em segmentos de tamanho variável. O Amoeba suporta objetos compartilhados que estão replicados em todas as máquinas que estão usando-os. Os objetos podem ser de qualquer tamanho e podem suportar qualquer operação.
- O modelo de memória do Mach (similar no Chorus) é baseado em objetos de memória e é implementado em páginas de tamanhos fixos. Quando uma thread referencia uma página que não está presente na sua máquina, a página é buscada na máquina em que se encontra e trazida para a máquina atual.

2) Cite algumas características do gerenciamento de memória no sistema Inferno.

O gerenciamento de memória está intimamente ligado ao conjunto de instruções da máquina virtual. O sistema pode executar sem uma MMU própria. A máquina virtual fornece gerenciamento de memória projetada para ser eficiente em dispositivos com o mínimo de 1 MB de memória e sem mapeamento de memória no hardware.

Questões sobre AMOEBA

P: Como podemos modelar didaticamente a divisão de sistema com amoeba com relação as funcionalidades locais e as funcionalidades remotas?

R: A tabela a seguir apresenta a divisão das funcionalidades no amoeba. As tarefas mais básicas são realizadas localmente em um microkernel. As funcionalidades mais elaboradas são responsabilidade de servidores específicos na rede

Microkernel(Local)	Servidores (Remoto)
1 gerenciar processos e threads	Servidor Bala
2 suporte para gerência de memória em baixo nível	Servidor de diretório
3 suporte a comunicação	Servidor de replicação
4 Manejar I/O de baixo nível	Servidor de processamento
	Servidor de boot

P:Qual o protocolo utilizado na comunicação com os servidores do sistema amoeba? Liste ao menos três servidores do sistema amoeba e suas respectivas funcionalidades

R:

Protocolo FLIP (*Fast Local Internet Protocol*)

Servidor bala: Principal servidor, gerencia armazenamento de informações em arquivos

Servidor de diretório: responsável pela nomeação. Fornece o mapeamento do nomes de arquivos para as respectivas capacidades

Servidor Replicação: atua sobre entradas de diretório que possuem menos capacidades do que deveriam ter. Responsavel por criar as copias, dado que o servidor de diretório apenas cria uma copia. Tambem implementa um sistema de envelhecimento dos objetos e de coleta de lixo.

Servidor de processamento: determina em qual arquitetura o processo deve ser rodado e gerencia o pool de processadores, designando o processador, ou conjunto de processadores que devem realizar a tarefa

Servidor de boot: Fornece tolerância a falhas, uma vez que verifica se os demais servidores estão funcionando. Caso algum não esteja, providencia uma solução

Servidor TCP/IP: Utilizado para comunicações em que não é possível utilizar o protocolo FLIP

Questões sobre IPC, Shared Memory, ...

(seguinto mais ou menos o padrão das perguntas sobre comunicação)

1 - Basicamente, como é feita a comunicação no Mach?

R - A base de toda comunicação no Mach são as chamadas portas. A thread que quer se comunicar escreve (se tiver permissão) uma mensagem na porta da qual a outra thread (a interlocutora) lerá. Como as portas são unidirecionais, é necessário que, se a thread leitora quiser responder, exista (ou seja criada) outra porta de onde a thread que começou a comunicação esteja leia.

2 - Como funciona a comunicação entre processos no DCE?

R - A comunicação se baseia em RPC. A camada de abstração do DCE relacionada ao RPC esconde do programador toda a preocupação sobre como as mensagens devem ser codificadas de forma que cheguem ao outro computador de maneira consistente. Isso permite que os programas cliente e servidor sejam escritos, inclusive, em linguagens diferentes.