Perguntas e Respostas – GVGO – Grupo 1 – Sistemas Distribuídos Clássicos e Reais

Questão 1
No Amoeba, os serviços e a comnicação entre eles são baseados em dois conceitos fundamentais. Quais são e como eles funcionam?

Resposta: Objects e Capabilities. Um object pode ser visto como um tipo abstrato de dados sobre o qual é possíel realizar operações (um diretório, um arquivo, etc). Cada objeto é gerenciado por um processo servidor para o qual é possível enviar mensagens RPCs (especificando o objeto, a operação e os parâmetros). Quando um objeto é criado, o servidor cria um valor criptografado de 128 bits chamado Capability e o retorna para o serviço que o chamou. Para realizar qualquer operação sobre o objeto criado é necessário enviar este Capability para o servidor, com o objetivo de identificar o objeto e assegurar que o usuario possui permissão para modificá-lo.

Questão 2
Qual a característica do DCE(Distributed Computing Environment) que faz com que seja fácil produzir software portável(roda em uma grande variedade de plataformas)?

Resposta: O DCE é um ambiente que roda em cima dos sistemas operacionais, permitindo que um usuário facilmente instale o software DCE em um conjunto de máquinas heterogeneo(diferentes tipos de hardware e sistemas operacionais) e rode aplicações distribuidas sem prejudicar outras aplicações não distribuidas existentes em cada máquina. O DCE roda em vários tipos de computadores e sistemas operacionais facilitando a produção de software distribuido portável que roda em uma grande variedade de plataformas, amortizando custos de desenvolvimento.

Questão 3
No sistema operacional distribuído E1, quais são as duas característica mais marcantes que, inclusive, identificam a arquitetura como não monolítica e orientada a objetos?

Resposta: Na implementação da abstração Single System Image, baseada na idéia de Objetos Distribuídos, as aplicações (e recursos) são construídas como uma coleção de objetos distribuídos que encapsulam um conjunto de interfaces que proporcionam uma visão do sistema distribuído como um sistema centralizado em todos os nodos - incluindo recursos de hardware. Isso garante ao programador uma visão do sistema como um único computador virtual (abstraindo seu layout físico) e viabiliza acesso a recursos através de métodos nos objetos. Observamos também a característica da replicação de objetos, na qual uma cópia (parcial ou total) de um objeto pode ser colocada em cada nodo em que está sendo usada. Cada invocação a um método de um objeto é tratada pela réplica do nodo em que a chamada foi originada e é responsabilidade das cópias comunicarem-se entre si quando, por exemplo, é necessário

sincronizarem-se ou obter alguma parte do estado do objeto que não possuam (dados danificados ou não replicados, por exemplo). A eficiência do acesso a um determinado objeto é determinado pela eficiência do conjunto de estratégias do protocolo de replicação, já que a comunicação distribuída no sistema está dentro dos objetos distribuídos.

Questão 4

Em relação ao microkernel Mach, como funciona a abstração de ports? Cite algumas das principais vantagens e desvantagens de se usar um esquema desse tipo para a comunicação com o kernel

Resposta: Uma porta (port) no Mach, representa basicamente um mailbox com proteção implícita. Essas estruturas suportam comunicação unidirecional entre 2 threads distintas, ou seja, quando uma thread X quer se comunicar com uma thread Y, X escreve uma mensagem para uma determinada porta e somente threads autorizadas recebem a mensagem, nesse caso Y.
O Mach utiliza esse esquema para minimizar o número de chamadas de sistema feitas ao seu microkernel, tentando sempre que possível utilizar portas para a comunicação com o kernel. Uma das principais vantagens dessa abordagem é a facilidade e a modularização do kernel, uma vez que o kernel fica com pouca participação, só dando acesso a porta levando ao programa que atende ao pedido. Uma desvantagem é que, devido a necessidade de garantir que as portas sejam seguras e protegidas, uma perda de performance considerável aconteça, uma vez que acarreta em uma verificação da validade e autenticidade das mensagens que passam por cada porta.

Questão 5

Como funciona a autenticação no ambiente DCE? Cite algumas ferramentas/algoritmos utilizados.

Resposta: Existe um servidor específico para autenticação, que realiza as operações:
- Identificação e autorização de usuários
- Autorização para aplicações decidirem se um usuário pode acessar uma operação ou objeto
- Comunicação segura para evitar eavesdropping

O servidor utiliza Kerberos para a identificação e autorização, fornecendo um 'ticket granting ticket' (TGT) para que o usuário possa usar a autoridade dele (servidor) para requisitar serviços, de acordo com os privilégios previamente configurados.

A proteção contra eavesdroppign é feita com DES, e a integridade é garantida com MD5.

Questão 6

Como são definidos os processoe no CHORUS? Como eles se comunicam? Quais as duas operações definidas para a comunicação?

Resposta: Um processo no Chorus é uma coleção de elementos ativos (threads) e passivos (espaços de endereçamento contendo algumas regiões, portas para trocas de mensagens) que trabalham conjuntamente na execução de uma computação. Todo processo ativo no Chorus tem uma ou mais threads, que executam código. Uma thread está amarrada ao processo em que foi criada, e não pode ser removida. Cada thread tem seu próprio contexto (pilha, registradores, etc.) que é salvo quando a thread é bloqueada à espera de algum evento e restabelecido quando a thread retoma o processamento.
A comunicação no Chorus ocorre à base de troca de mensagens. Cada mensagem contém um cabeçalho, para uso do microkernel, uma parte fixa opcional para controle do usuário, de 64 bytes, e um corpo de no máximo 64 Kbytes, também inteiramente destinado ao usuário.
Chorus oferece duas operações de comunicação: envio assíncrono e RPC.
Envio assíncrono permite que uma thread simplesmente envie uma mensagem a uma porta, sem garantia de que a mensagem chegará ao destino e em notificação caso algum problema ocorra.
Quando um processo executa uma operação RPC, fica bloqueado, sendo desbloqueado quando recebe a resposta ou por timeout. RPC usa a semântica at-mostonce, ou seja, a solicitação para um mesmo serviço é feita no máximo uma vez. É possível mandar uma mensagem para um grupo de portas.

Questão 7

Quais são os principais objetivos para a utilização do DCE?

Resposta: Um objetivo para utilizar o DCE são utilizar máquinas com sistemas operacionais existentes, necessitando apenas a instalação do DCE software para poder rodar as aplicações distribuídos.

Outro objetivo é a maior facilidade no desenvolvimento de sistemas distribuídos portáveis por não necessitar de máquinas e sistemas operacionais iguais, apenas necessitando a instalação do DCE. E por tornar algumas conversões de dados transparentes ao programador.

Podemos também considerar como uma grande motivação para utilizarmos o DCE, a sua biblioteca de serviços de segurança já implementadas que entre outras atividades, provê autenticação de máquinas clientes e criptografia dos dados que trafegam via RPC

Questão 8

Explique como funciona o escalonamento de threads no Mach especificamente para um sistema com multiprocessador e o que significa 'handoff schedulling'.

Resposta: Como o Mach foi projetado para funcionar em máquinas com um multiprocessador, existe o conceito de conjunto de processadores, na qual os processadores do sistema podem ser agrupados em conjuntos disjuntos de processadores, por software. As threads podem ser designadas para um conjunto de processadores ou para um processador específico, por software. O escalonamento é baseado em prioridades, que vão de 0(maior prioridade) até 31 ou 127(menor prioridade). Cada processador e conjunto de processadores possui uma fila, ordenada por prioridades, chamada local e global, respectivamente. A fila global contém threads que podem ser designadas a qualquer processador do conjunto.

O algoritmo de escalonamento verifica inicialmente na fila local qual a thread de maior prioridade, se ela estiver vazia, é feita uma busca na fila global, estando ela vazia, uma thread ociosa é escolhida. A cada tick do relógio, a CPU incrementa o contador de prioridade da thread (diminuindo sua prioridade). Cada thread roda por um determinado quantum, que é variável e depende do número de threads aptas a rodar.
Handoff schedulling é um mecanismo que permite a uma thread nomear qual a próxima thread será escalonada numa CPU, passando diretamente sem o uso do escalonador. Se usado inteligentemente pode aumentar a performance, o kernel o usa em algumas circunstâncias como otimização.

Questão 9
Qual é o modelo de sistema implementado no AMOEBA e como ele funciona?

Resposta: o AMOEBA utiliza o sistema "processor pool", que consiste em um rack cheio de CPU's que podem ser dinamicamente alocadas sob demanda. Usuários podem requisitar quantos processadores eles precisarem por curtos períodos, sendo que depois de usados os processadores são retornados ao "pool". Nao existe conceito de posse nesse modelo, todos CPU's pertencem à todos, e se nao houverem processadores suficientes, novos processos serão automaticamente alocados às CPU's menos carregadas.

Questão 10
Quais os tipos de gerenciamento de memória oferecidos pelo Chorus?

Resposta: O Chorus oferece diversos serviços que permitem extender seu espaço de endereçamento dinamicamente, alocando regiões de memória. É possível também diminuir seu espaço de endereçamento liberando regiões de memória. O Chorus permite o compratilhamento de uma área de memória entre dois ou mais processos, sejam eles de usuário ou supervisores.
Há três modelos de gerenciamento disponíveis: flat - microkernel e aplicações rodam no memsmo espaço de endereçamento, sem proteção -, protegido - espaços separados, usado quando proteção de memória é imprescindível - e virtual - quando é necessário mais memória que o disponível fisicamente.

Questão 11
Explique como funcionam as operações de write no sistemas de arquivos do Amoeba e do DCE.

Resposta: No Amoeba, não é possível escrever dados em arquivos já existentes, pois os arquivos são imutáveis. Para isso, o cliente deve receber o arquivo (mandando a capability correspondente ao sistema de arquivos), fazer as modificações localmente e então enviar esse novo arquivo ao sistema como um novo arquivo, com uma nova capability. O sistema de arquivos então decidirá se irá deletar o antigo arquivo ou mantê-lo como back-up.
No DCE, as operações de write, assim como a maioria das outras operações de arquivo, são feitas com o auxílio de tokens. Quando um cliente deseja escrever num arquivo, ele recebe um token referente a esse arquivo. Se um outro cliente deseja realizar a mesma operação, ele não poderá fazê-lo, pois o servidor não possui mais o token daquele arquivo. Assim, o servidor pede novamente o token ao primeiro cliente, que o devolve quando for conveniente, e o repassa para o novo cliente.

Questão 12
Discuta sobre as principais razões pela qual o modelo de kernel NT (utilizado pelo Windows NT) não é considerado um microkernel.

Resposta: Um microkernel, por definição, é um tipo de kernel que não contém nenhum serviço que os Sistemas Operacionais devem oferecer, apenas a plataforma para esses serviços. Desta maneira um microkernel só deve conter as funcionalidades mais básicas, como gerência de memória, gerência do processador, IPC (comunicação entre processos), suporte à multithreading, etc. Os serviços do OS serão, então, fornecidos por servidores que rodam com privilégios de usuário, em uma região da memória diferente da do kernel. Partindo deste conceito, o NT não segue a risca estas características, apesar de seus obejtivos de design serem de um microkernel. A maioria de seus subsistemas são implementados dentro da área de endereçamento do kernel, executando em modo privilegiado, como por exemplo o acesso aos dispositivos de hardware, o sistema de janelas, etc. Isto traz benefícios de performance, já que os subsitemas não precisam utilizar IPC para se comunicar, tirando o overhead que isto implicaria (como no caso do Mach).

Questão 13
Como o kernel pode ter tolerancia a falhas num sistema distribuido? Quais os problemas recorrentes disso?

Resposta: O kernel pode ser tolerante a falhas utilizando a replicação de dados e processos. Porém, garantir dependabilidade envolve solucionar problemas de consenso, ordenação e atomicidade na troca de mensagens entre grupos de processos, sincronizar relógios quando necessário, implementar réplicas consistentes de objetos, garantir resiliência de dados e processos num ambiente sujeito a quedas de estações tanto clientes como servidoras, particionamento de redes, perda e atrasos de mensagens e, eventualmente, comportamento arbitrário dos componentes do sistema. Isto torna difícil a gerência do sistema, podendo resultar em perda de performance.

Questão 14
Todos os recursos do sistema Inferno, tanto locais quanto remotos, são representados por um conjunto de arquivos dinâmicos dentro de um sistema hierárquico de arquivos. Entre eles, recursos de dispositivos de armazenamento, processos, serviços, redes e conexões de rede. Quais as vantagens de se usar arquivos como conceito central do sistema?

Resposta: Uma aplicação pode acessar cada recurso manipulando os arquivos relevantes usando simples operações de aquivos. As vantagens disso são:
* Arquivos de sistemas têm interfaces simples e intuitivas em uma grande variedade de sistemas operacionais. Interfaces em arquivos consitem num pequeno conjunto de operações bem definidas como "abrir", "ler" e "escrever".
* A confiança em sistemas de arquivos reduzem a quantidade de código das interfaces e mantém o sistema Inferno pequeno, confiável e altamente portável.
* Convenção para nomes de arquivos são bem conhecidas, uniformes e facilmente entendíveis.
* Direitos de acesso e permissões a arquivos são simples e ainda podem ser usadas para assegurar múltiplos níveis de segurança.

Questão 15
Qual a diferença das implementações de memória compartilhada e distribuída entre o Chorus, o Amoeba e o Mach?

Resposta: O Amoeba suporta objetos (tipos encapsulados de dados = variáveis compartilhadas mais métodos) compartilhados que são replicados em todas as máquinas que os utilizam, pois o Amoeba não suporta paginação, sendo de tamanhos constantes. Apesar disso, esses objetos podem possuir qualquer tamanho e quaisquer operações. As leituras são locais e escritas realizadas através do protocolo de broadcast.
O Mach e o Chorus suportam memória compartilhada e distribuída baseada em paginação. Quando uma thread busca uma página que não está em sua máquina, ela precisa ser buscada na máquina que a possui, sendo emitida uma mensagem no barramento.
Por fim, a implementação do Amoeba é mais cara devido à replicação de objetos, mas previne os potenciais thrashings que podem ocorrer no Mach e no Chorus.

Questão 16
Explique como se dá a forma de comunicação por grupo do Amoeba.

Resposta: atrave´s do chamado “Protocolo de broadcast confiável”, onde acontece o seguinte:

1) Processo usuário faz uma chamada de sistema para o kernel, passando a mensagem;
2) Kernel aceita a mensagem e bloqueia o processo de usuário;
3) Kernel envia uma mensagem ponto-a-ponto no seqüenciador;
4) Após pegar a mensagem, o seqüenciador atribui um número ao pacote e envia a mensagem ao grupo todo. Esse número é gerado por uma seqüência, de forma a controlar a seqüencialidade de envio dos pacotes;
5) Quando o kernel vê que a mensagem foi enviada, ele desbloqueia o processo de usuário e o deixa continuar sua execução.

Através dessa seqüencialidade de envio dos pacotes, o seqüenciador mantém a consistência das entregas dos pacotes. Por exemplo, quando o seqüenciador envia um pacote numerado como 5, após o envio cada receptor verifica se o pacote recebido anteriormente estava numerado como 4. Caso um receptor não tenha recebido esse pacote (de número 4), ele faz uma requisição ao seqüenciador para receber esse pacote e o seqüenciador, através de um repositório, reenvia o pacote 4 para o requisitor. Como o buffer do seqüenciador é limitado, quando é atingido um limite de envio de pacotes, digamos, de 1 até 25, ele pergunta aos membros do grupo se todos têm os pacotes numerados de 1 a 25. Caso falte um deles para um membro do grupo, o seqüenciador reenvia esse pacote, esvaziando o buffer logo em seguida.