GVGO – 2012/02 - Grupo 1

Integrantes:

Glauber Hermany (Lotus)
João Gross (Amoeba)
Hélio Brauner (Amoeba)
Guilherme Bender (Amoeba)
Guilherme Nunes Serafina (DCE)
Jonas Crauss (DCE)
Gustavo Miotto (Mach)
Luis Leiva (Mach)
Nicolas Kagami (Chorus)
Luís Reinicke (Chorus)
Álan Dias (Chorus)
Guilherme Schievelbien (Inferno)
Vinícius Graciolli (Inferno)

(Mach) Questão 1: Entre as propriedades mais importantes de cada processo, são necessários o conjunto de threads e o espaço de endereçamento, e mais quatro portas essenciais, quais são essas portas e o que elas fazem?

- **process port**: é usada para fazer a comunicação do kernel com o processo. Muitos dos serviços do kernel são requisitados por essa porta apenas mandando uma mensagem a ela, em vez de fazer uma chamada de sistema. Isso reduz o número de "system calls" a uma quantidade bastante pequena.

- **bootstrap port**: realiza a inicialização do processo quando esse é carregado. Através dessa porta o processo fica sabendo o nome das portas do sistema, e Processos UNIX também a utilizam para se comunicar como emulador UNIX, no caso de processos UNIX emulados.

- **exception port**: é usada para reportar exceções causadas pelo processo. Exceções típicas são divisão por zero e a execução ilegal de instruções. A porta informa ao sistema para onde a mensagem de exceção deve ser enviada. Programas de depuração também usam essa porta para realizar seu trabalho.

- **registered ports**: são portas normalmente associadas a processos padrão do sistema, proporciona um meio para o processo se comunicar com servidores de sistema padrão. Como o servidor de nomes, por exemplo.

(Mach) Questão 2: O Compartilhamento entre threads no MACH é automático, ou seja, não precisa ser programado, mas o compartilhamento de memoria entre processos não, cite como funcionam os três tipos de compartilhamento de memória entre processos pais e filhos que o MACH fornece?

- **A região não pode ser usada pelo processo filho**: indica que o processo pai não quer compartilhar aquela região e qualquer acesso a ela será tratado como um acesso a uma área não alocada.

- **A região é compartilhada**: o compartilhamento acontece normalmente, e qualquer modificação em uma cópia acarreta na atualização da outra.

- **A região no filho é uma cópia da do pai**: a região é copiada e qualquer modificação será feita somente na cópia local. Essa opção é implementada utilizando-se um artifício de copiar-na-escrita, ou seja, a cópia física somente é realizada se houver alguma modificação na cópia. Isso economiza preciosos ciclos de CPU no caso de regiões que nunca são acessadas.

(Mach) Questão 3: No sistema Mach temos associado a cada conjunto de processadores um array de filas de execução. Quantas filas cada array possui e, quantas e quais são as variáveis associadas a cada fila?

Cada array possui 32 filas que correspondem a threads com prioridades de 0 a 31, quando uma thread de prioridade n se torna executável ela é colocada no final da fila n; se uma thread não é executável ela não se encontra em nenhuma fila de execução.

Cada fila de execução possui três variáveis associadas a ela, a primeira é um mutex utilizado para bloquear a estrutura de dados e é usado para garantir que somente um processador por vez está manipulando as filas; a segunda é um contador referente ao número de threads em todas as filas combinadas, se é zero não há trabalho a ser feito; a terceira é uma "pista" de onde encontrar a thread com maior prioridade.

(DCE) Questão 4: Como o cliente conhece os métodos que podem ser invocados no servidor?

É necessário criar uma interface entre esses dois antes da compilação do programa. Para isso, é necessário o arquivo "*interface definiton file*", escrito na linguagem IDL (*interface definiton language*). Esse arquivo deve especificar todas as funções que podem ser invocadas no servidor, assim como seus parâmetros e valores de retorno. Vale lembrar que cada arquivo de interface é gerado com um identificador único, um número de 128 bits. O cliente deve enviar esse identificador na primeira troca de mensagens por RPC para informar através de qual interface está se comunicando. Caso o servidor não reconheça esse número, a requisição é desprezada.

O *interface definition file* é compilado e gera um arquivo de header que deve ser incluído (*#include*) tanto no servidor como no cliente. Esse arquivo possui os protótipos das funções, definições de constantes e o identificador único da interface. Além do header ele gera arquivos stub tanto para o cliente como para o servidor. Esses arquivos stub devem ser compilados junto com o código do programa que irá rodar na máquina local. No caso do cliente, esse arquivo é responsável por empacotar os parâmetros e chamar a função de sistema correspondente. No caso do servidor, esse arquivo desempacota os parâmetros e chama a função local correspondente.

(DCE) Questão 5: Como funciona o conceito de célula no DCE?

Em um sistema DCE, usuários, máquinas e outros recursos são agrupados em células. Vários aspectos do DCE são baseados nelas e a divisão das células normalmente reflete a organização da companhia que está usando o sistema. Isso faz com que a determinação de como o sistema vai ser dividido em células tende a levar em conta mais questões de negócios do que questões técnicas do sistema. Quatro principais fatores são levados em conta para determinar como agrupar as máquinas e recursos em células:

1. Propósito: uma vez que as máquinas de uma mesma célula costumam partilhar um mesmo objetivo global à célula ou oferecer um mesmo recurso.
2. Segurança: tem a ver com o fato de que o DCE funciona melhor quando os usuários de uma célula confiam mais em outros usuários dessa mesma célula do que em usuários de outra célula. Isso por que os limites da célula agem como *firewalls* e a comunico entre células é trabalhosa.
3. Sobrecarga: é importante por que algumas funções do DCE são otimizadas para funcionar dentro das células. Quando se coloca duas máquinas distantes em uma mesma célula é ruim, por que elas vão ter que comunicar via WAN.
4. Administração: toda célula precisa de um administrador.

O tamanho das células pode variar muito, no entanto, todas as células devem ter um servidor de tempo, um de diretório e um de segurança. Também, deve haver pelo menos um cliente ou um servidor.

(LOTUS) Questão 6: No sistema LOTUS existem 3 funções lógicas em um acesso a arquivos e assim existem 3 sites logicos. Quais são os diferentes sites e suas respectivas funções?

a. using site, (US), que faz requisição de abertura de arquivo e ao qual as páginas dos arquivos devem ser fornecidas.

b. storage site, (SS), que é o site onde uma cópia do arquivo solicitado esta armazenado e que foi escolhido para fornecer páginas do arquivo para o using site (US)

c. current synchronization site, (CSS), que reforça uma política de acesso global sincronizado para os arquivos do grupo de arquivos e seleciona os SS para cada requisição aberta. Um site físico pode ser o CSS para qualquer número de grupo de arquivos mas há somente um CSS para um dado grupo de arquivos em qualquer conjunto de sites que comunicam-se. O CSS não precisa armazenar qualquer arquivo em particular no grupo de arquivos mas a fim de que tome decisões precisas de acesso ele deve ter conhecimento de quais sites armazenam o arquivo e qual é a versão mais recente do arquivo.

(Chorus) Questão 7: Descreva a estrutura de elementos que compõem os processos do Chorus.

Os processos do chorus são compostos de elementos ativos, as threads, e elementos passivos, a memória compartilhada e as portas para comunicação entre processos.

Threads: Cada thread possui um contexto que inclui a área de pilha, registradores e não pode ser removida de um processo após criada. A implementação inclui prioridades, threads prioritárias só deixam a CPU se terminarem ou se bloquearem, as outras threads são escalonadas por round-robin, implementado com mutexes e semáforos.

Memória: Memória estruturada em páginas, com o acesso de leitura permitido a todos, mas acesso exclusivo de escrita.

Portas: A comunicação pode ocorrer de duas formas, envio assíncrono e RPC. No envio assíncrono uma mensagem é enviada sem garantia de entrega e de forma não-bloqueante. O RPC por outro lado, é bloqueante e só volta após a reposta de

execução do RPC (ou por um timeout) e usa a política at-most-once. Uma mensagem pode ser enviada para múltiplas portas simultaneamente.

Existem mini-portas que permitem mini-mensagens que são usadas entre processos do kernel para sinalizar interrupções.

(Chorus) Questão 8: Quais os tipos de troca de mensagem no Chorus? Como essas mensagens podem ser enviadas?

Mensagens assincronas e RPC. Podem ser envidas:

1 - Para todos os membros;
2 - Para qualquer membro;
3 - Para um membro específico;
4 - Para qualquer membro menos um específico.

(Inferno) Questão 9: Como o Inferno diferencia processos de arquivos?

Ele não diferencia, todos os recursos são vistos como arquivos no Inferno.

(Inferno) Questão 10: Qual a vantagem de ver todos os recursos como arquivos?

Isso garante uniformidade dos comandos de acesso.

(Amoeba) Questão 11: Explique de forma sucinta como funciona o sistema de "capabilities" do Amoeba para o controle das operações que um usuário pode realizar.

Uma "capability" é um um valor (de 128 bits) construido pelo servidor cada vez que um objeto é criado, e enviado para quem fez a chamada que originou o objeto. Qualquer operação que seja requisitada sobre este objeto exige que o usuário envie essa "capability" para o servidor, permitindo especificar o objeto a ser manipulado e garantir que este usuário possui permissão para manipular o objeto especificado.

(Amoeba) Questão 12: Qual é o objetivo do sistema Amoeba?

O Amoeba é um sistema de timesharing que faz toda a rede de computadores aparecer ao usuário como se fosse uma só máquina, ou seja, o usuário não sabe qual máquina que está usando, pois para ele isso é transparente. Todos os recursos do sistema ficm disponíveis até que sejm requisitados para rodar os processos dos usuários. A partir dessa requisição os recursos são alocados para os processos dos usuários e o processos são executados.

No caso de todos os recursos estarem em uso, é aplicada a divisão de tempo da cpu, ou seja, cada usuário tem disponível uma fati de tempo para rodar seus processos. Dessa forma há um melhor aproveitamento dos recursos e justiça na execução para todos os usuários.