**Comunicação**

1. Escolha três padrões de rede usados em aplicações embarcadas e compare-os segundo os requisitos: Bandwidth, Robustez, Facilidade de reparos, Privacidade e Tempo-real

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Bandwidth | Robustez | Facilidade de reparos | Privacidade | Tempo-real |
| Bluetooth | Até 1Mbit/s | Bom, salto de frequência | Não faz sentido, meio aéreo | Fácil de interceptar | Não é bom |
| CAN | Até 1Mbit/s | Bom, par trançado | Fácil | Não faz sentido | Bom, CSMA CA |
| Ethernet | Até 10Mb/s | Bom, par trançado | Fácil | Fácil de interceptar | Não é bom, CSMA CD |

**2. Explique a diferença entre comunicação orientada a conexão e publish-subscribe. Dê um exemplo de aplicação embarcada para cada uma.**

A conexão orientada a conexão se preocupa em estabelecer uma conexão entre os pontos comunicantes, além de ser uma comunicação ponto a ponto. A comunicação publish-subscribe não tem conexão entre pontos, apenas nós que “publicam” os dados em um barramento e nós interessados nesses dados “são inscritos” e podem recuperá-los. Quando o publicador envia dados ele não envia a nenhum nó em específico.

**3. Descreva o barramento I2C dando os detalhes para os níveis físico, enlace, rede e transporte do modelo OSI.**

No nível físico o I2C utiliza 2 fios, um para transmissão do clock e o outro para transmissão dos dados. Além disso o barramento utiliza o esquema de coletor/dreno aberto.

No nível de enlace cada nó tem um endereço que é utilizado para identificar unicamente o nó na rede.

No nível de rede e transporte eu não sei.

4. Considere como aplicação um sistema de elevadores, como mostrado na figura abaixo. Cada andar possui painel de controle de displays. Cada elevador possui um botão por andar e botão de emergência. Sensores indicam a posição do elevador por todo o prédio, com aumento da resolução na vizinhança dos pontos de parada. Existe um controlador central (monitora posição dos elevadores e chamados nos andares, despacha elevadores para atender chamados) e um controlador em cada elevador (controle de movimento).

a) Proponha uma arquitetura de rede.

b) Selecione uma tecnologia de rede.

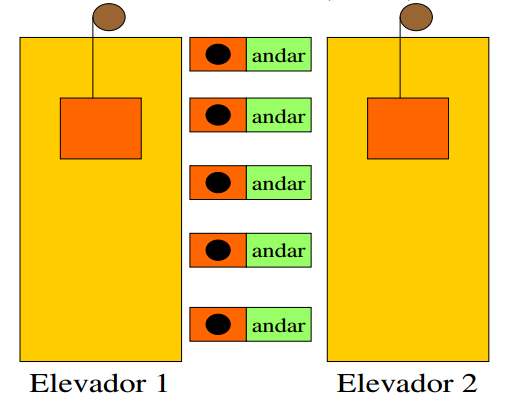
c) Que propriedades seriam necessários nos níveis: físico, enlace, redes e transporte desta rede?

Físico: proteção contra interferência analógica das outras partes do elevador.

Enlace:

Rede:

Transporte:



**5. Como o protocolo CSMA-AMP pode contribuir para aumentar o determinismo de uma rede? Comente o efeito sobre os nós que possuem baixa prioridade.**

Ele determina que os nós com maior prioridade terão o barramento garantido, com isso sabe-se exatamente quem dominará o barramento. Os nós de baixa prioridade podem nunca ganhar o barramento e sofrer “starvation” se muitos nós de alta prioridade acessarem o meio constantemente.

**6. A figura abaixo mostra um cabo USB. Por que os fios de dados são enrolados e os de alimentação não?**

Por que os fios de dados usam a técnica de par trançado e dados diferenciais para a eliminação de ruído e interferência, ou seja para proteger os dados. O cabo de alimentação não precisa de proteção contra corrupção de dados.



**7. Diferencie robustez de tolerância a falha.**

Robustez é a capacidade do sistema de funcionar mesmo quando submentido a situações físicas extremas. Tolerância a falha esta relacionado com a capacidade do sistema de continuar funcionando mesmo após ocorrer uma falha, ou seja, a sua capacidade de se recuperar dessas falhas.

8. Pesquise a largura de banda (bandwidth) das seguintes tecnologias de comunicação: Ethernet (1000BASE-LX e 100BASE-TX), Bluetooth (todos) e usb (2.0 e 3.0)

**9. Comente a robustez do padrão par trançando, utilizado nos cabos USB e ethernet (fio).**

Par trançado usa a técnica de transmissão diferencial que é menos suscetível a ruídos externos. Dessa forma em um ambiente com muito ruído o par trançado se sairia bem pela utilização dessa técnica. Onde um sinal e seu complementar são enviados em dois fios diferentes e no final existe um comparador que remove os ruídos adicionados ao longo do caminho.

**10. Compara os procedimentos de manutenção das tecnologias ethernet com fio versus ethernet fibra.**

Manutenabilidade em fibra ótica é mais difícil pois exige equipamentos extras.

**11. Compare as tecnologias ethernet wi-fi versus Bluetooth no quesito privacidade.**

Wifi, por ser mais utilizada para acesso a rede tem padrões e técnicas de proteção de dados mais avançadas do que o bluetooth que implementa pouco tais técnicas.

**12. Explique o padrão publish-subscribe. Como ele se diferencia do modelo usado no TCP/IP?**

No padrão publish subscribe existe um nó que envia dados para a rede, em estilo broadcast, ou seja sem um destinatário específico e existem nós que são assinantes de certos canais de recepção. Assim um publicante envia os dados para a rede e um assinante pega os dados que lhe interessam. No modelo TCP/IP existe um estabelecimento de conexão ponto a ponto.

**13. Pesquise sobre o protocolo CSMA NDA (Non-Destructive Arbitration) usado no CAN. Como ele consegue garantir que mensagens com maior prioridade sempre ganham o barramento?**

Através da arbitração prioritária do endereço. No período de arbitração todos os nós que desejam enviar dados tentarão enviar um pacote para rede, a primeira parte do pacote é o número de prioridade desse nó. Quando um nó de menos prioridade envia seu número de prioridade e outro nó de maior prioridade também envia o nó de menor prioridade consegue sentir a presença do outro nó, pois em certo momento ele tentará escrever 0 na rede, mas vai sentir um 1, indicando a presença de um nó com maior prioridade, assim ele irá parar de transmitir permitindo o nó de maior prioridade continue sua transmissão.

**SISTEMAS OPERACIONAIS**

**1. O que é device driver? Qual a sua importância no projeto de um sistema embarcado?**

É um pedaço de código que saber como se comunicar com o hardware. Um desenvolvedor de aplicações pode utilizar os device drivers para fazer chamadas ao hardware. Sua importância em sistemas embarcados está no seu papel de intermediário entre a aplicação e o hardware pode-se desenvolver o projeto em duas esferas e comunicando-as apenas através dessa interface.

**2. O que motivaria uma equipe de desenvolvimento a abandonar a estratégia de “laço simples” e partir para um “laço combinado com serviço de interrupção”?**

Quando os blocos de tarefas no laço simples se tornam muito longos ou quando um bloco não pode esperar que o outro termine seu serviço é mais interessante usar o serviço de interrupção para definir prioridades e ativar certos blocos nos momentos certos.

**3. O que motivaria uma equipe de desenvolvimento a abandonar a estratégia de “laço combinado com serviço de interrupção” e partir para o uso de um sistema operacional?**

Quando a quantidade de tarefas e serviços a serem implementados for muito grande tal que a administração disso de forma “manual” se torna incomodo. Ou quando existe uma necessidade de modularizar as tarefas de forma que elas possam ser desenvolvidas por indivíduos diferentes de um time de forma independente. O sistema operacional é capaz de administrar essas tarefas de forma melhor e atender a um número maior de tarefas.

**4. Qual das três estratégias seria melhor para o caso do elevador acima? Mudaria alguma coisa de fosse um sistema de 8 elevadores?**

Para um único elevador pode-se usar laço combinado com ISR pela quantidade baixa de subsistemas que precisam ser administrados.

Se aumentarmos para 8 elevadores e um único sistema fosse necessário por administrá-lo então seria melhor utilizar um SO por conta do número elevado de subsistemas para serem administrados.

**5. Que propriedades devem ser buscadas em um sistema operacional para uma aplicação embarcada que requer baixa dissipação de potência? Justifique.**

O sistema operacional deve ser eficiente, ou seja, deve utilizar apenas funções que são realmente necessárias. Se um SO tem uma boa configurabilidade então pode-se remover partes dele que não serão usadas sendo eficiente na memória o que afeta diretamente a potência. Deve ser eficiente no código da aplicação, novamente para diminuir a utilização de memória. Um SO mais simples/eficiente pode rodar sobre uma organização (CPU) mais simples o que também diminui a dissipação de potencia, ou ainda rodar a frequências de clock menores.

6. Escolha três sistemas operacionais usados em aplicações embarcadas e compare-os segundo os requisitos: custo de licenciamento, diversidade de plataformas, suporte, ferramentas de desenvolvimento e complexidade da aplicação.

**7. Qual o papel do middleware em uma aplicação embarcada?**

Simplesmente provê recursos que podem ser acessados como se fosse locais, mas que na verdade estão distribuídos em sistemas que podem rodar SOs e linguagem diferentes daqueles em que o sistema cliente está.

**LINGUAGENS**

**1. Considerando as linguagens Assembly, C e Java para desenvolvimento de aplicações embarcadas, identifique os tipos de aplicação mais adequados para cada uma delas. Sugira plataformas de hardware mais adequadas para cada uma e dê um exemplo de aplicação.**

Assembly é mais utilizado em bibliotecas ou aplicações mínimas. Intel 8051.

C é o mais utilizado no mundo embarcado hoje em dia. PICs

Java é usado quando um nível de complexidade da aplicação é muito grande e é necessário um nível de abstração que o C não proporciona. ARM9 Jazelle.

2. Caracterize as linguagens Assembly, C e Java para desenvolvimento de aplicações embarcadas em relação aos aspectos: facilidade de reuso, desempenho, e uso de memória.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Assembly | C | Java |
| Reuso | Difícil, existem poucos programadores Assembly. | Melhor que o Assembly, mas pior que o java | Reuso mais facilitado |
| Desempenho | Melhor desempenho pela proximaidade do hardware | Desempenho pior que o assembly, mas melhor que o do java. Embora dependendo do compilador o java pode se equiparar | Pior desempenho. |
| Memória | Uso mais efetivo da memória. | Segundo uso mais efetivo de memória. | Terceiro uso mais efetivo de memória, o POO pode ser um overhead nesse quesito. |

**3. Indique duas estratégias utilizadas por compiladores para diminuir o consumo de energia em aplicações embarcadas.**

Fazer o escalonamento das instruções baseado em energia, ou seja atribuir a cada instrução um ”valor” de energia gasta e escaloná-las de acordo com esse valor. Ou seja, manipula-se a ordem das instruções para reduzir os chaveamentos.

Alterar o critério de seleção de uma instrução(ou cadeia de instruções) que representa um código fonte para um critério baseado no gasto de energia.

**4. O que leva alguns desenvolvedores a utilizar código Assembly em partes de suas aplicações?**

Quando um desenvolvedor importa bibliotecas e as utiliza em seu código muitas vezes ele está utilizando códigos em assembly sem saber. As bibliotecas são feitas em assembly para alcançar os melhores resultados de desempenho. Um desenvolvedor pode escolher fazer parte do seu código em assembly caso queira alcançar um desempenho que uma linguagem compilada não conseguiria alcançar, para isso entretanto é necessário um conhecimento profundo da arquitetura alvo.