

Computadores

HW e SW

Design dos
Computadores

Requisitos e
Objetivos da Missão

1. Definir Requisitos

- Avaliar os objetivos que indicam sistemas computacionais.
- Avaliar requisitos de todas as disciplinas que indiquem utilidades de software
- Desenvolver uma arvore de requisitos de computação

- Avaliar arquiteturas e interfaces candidatas
- Realizar a divisão das funções
- Avaliar requisitos de confiabilidade
- Estabelecer um baseline

2. Alocar os requisitos de
alto nível

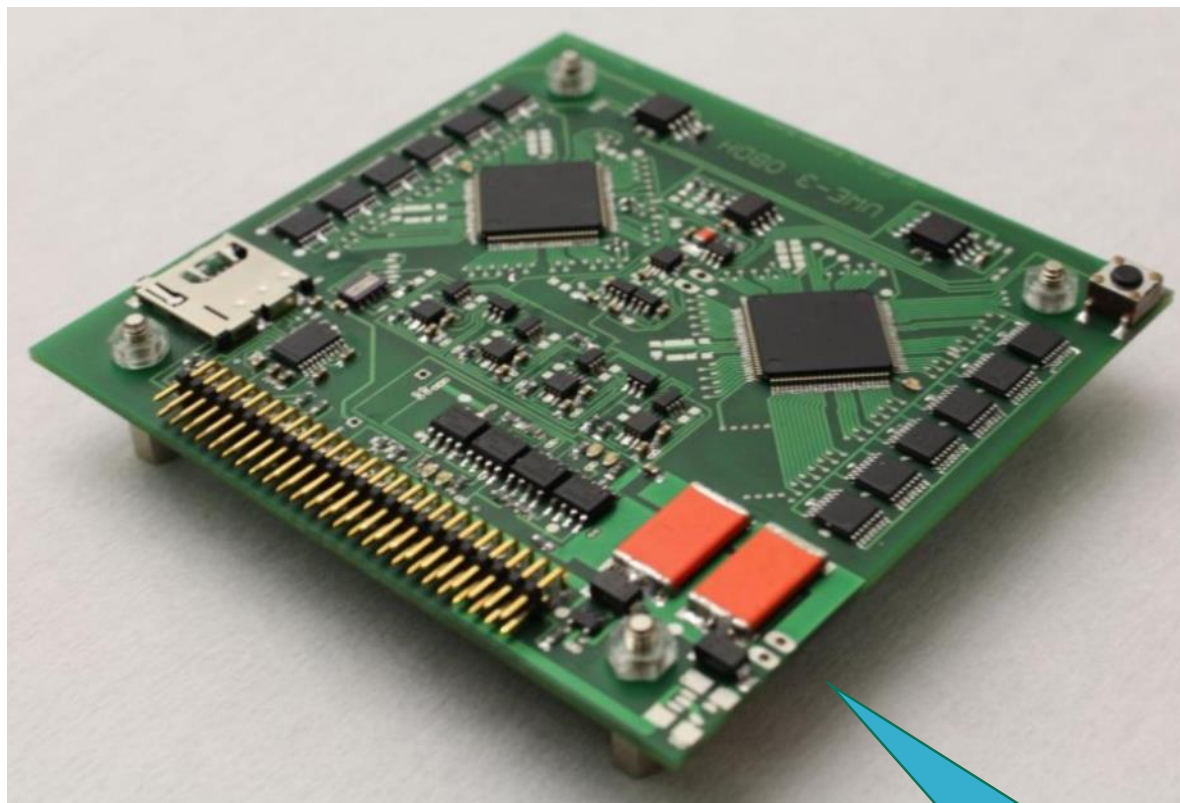
3. Definir os requisitos do
Computador

- Definir as tarefas
- Estabelecer os requisitos de temporização e tamanho do software
- Estabelecer os requisitos de tamanho, peso, consumo e processamento do Computador
- Identificar Hardwares possíveis

- Identificar o sistema operacional e o ambiente de desenvolvimento
- Identificar as ferramentas de suporte
- Detalhar o processo de controle de versão/configuração
- Documentar os testes e abordagens de integração
- Deploy (entrega) do sistema, e o procedimento de manutenção

4. Definir a estratégia de
desenvolvimento e
entrega

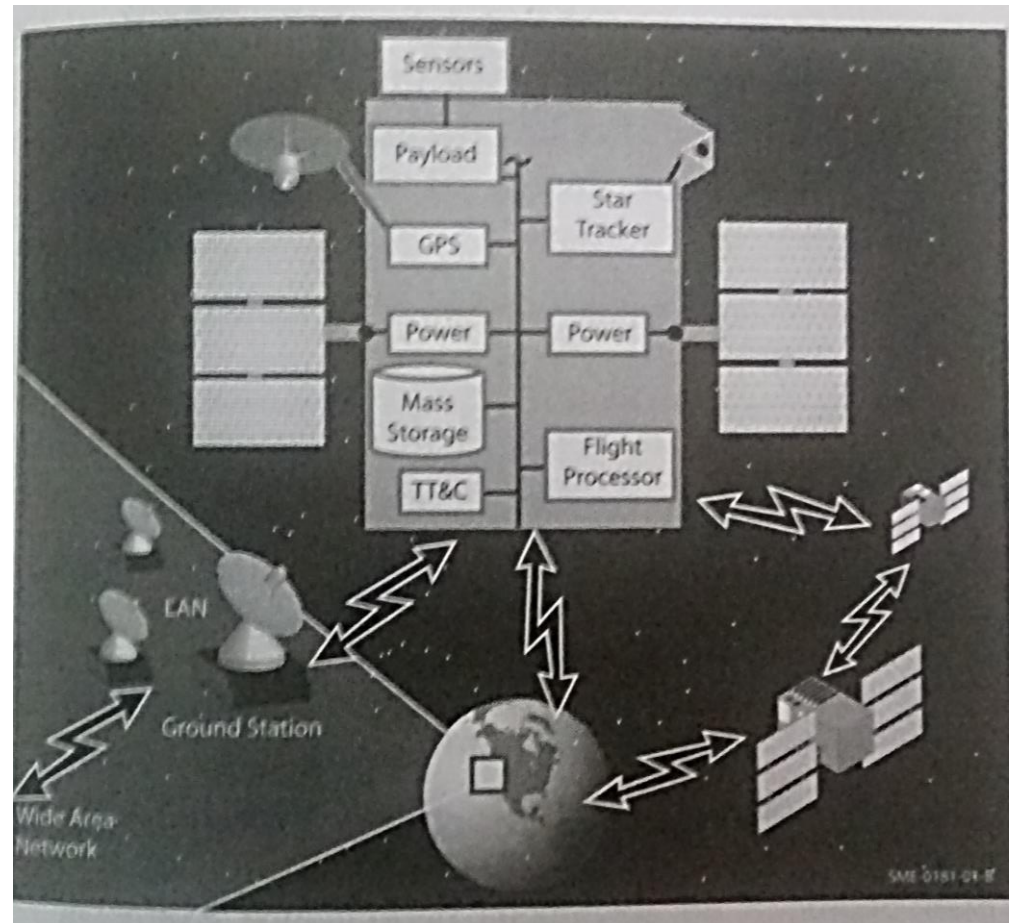
OBDH – On Board Data Handling



- Redundância (cold/hot)
- Barramentos de dados
- Tipos de Interfaces
- Relógio
- Processamento de Telecomandos
- Encapsulamento de telemetrias
- Armazenamento
 - Software
 - Coleta de dados

Aqui tem Software Embarcado

- Os sistemas computacionais que suportam a missão incluem o computador de bordo, os computadores auxiliares (Cargas-úteis / Controles), bem como os computadores de solo.



Conceitos

- **Sistemas Embarcados (*Embedded Systems*):**

- Um processador ou microprocessador, provendo controle em tempo real, como um dos componentes de um sistema maior.

- **Processamento em Tempo Real (*Real-Time Processing*):**

- Manuseando, e processando informação no tempo que o evento ocorre ou quando a informação é criada.

- ***Hard Real-Time:***

- Requisito que a precisão de tempo precisa ser ótima, senão causa severas consequências.

- ***Soft Real-Time:***

- Requisito que a precisão de tempo pode desviar sem causar consequências.

- **Software do Sistema Operacional (*Operational System Software*):**

- Gerencia os recursos do computador e as atividades que precisam dos recursos.

- **Software da Aplicação em Voo (*Application Flight Software*):**

- Software específico da missão que contém o trabalho requerido pelo usuário ou pela missão.

... Informações... O que precisa existir:

- Req. de Sistema → Alocar em Subistema → identificar modos de operação → alocar requisitos de alto nível do Computador de bordo → definir interface com outros subsistemas → identificar requisitos periféricos,

HW

- Itens de configuração de HW
- Placa
- Chips / Componentes / Interfaces

SW

- Itens de configuração de SW
- Componentes de Software
- Unidades de software (onde vai ser processado/entregue)

DOC

- Requisitos
- Design
- Design Detalhado
- Controles de Interface (Interface Control Documents – ICDs)

Para chegar num baseline devemos:

- 1. Alocar os requisitos de missão e sistemas em componentes de processamento (funções)
- 2. Particionar e alocar os requisitos em espaço/solo, carga-útil/barramento, subsistema, hw/sw
- 3. Definir os requisitos do sistema computacional, detalhando o tipo de processamento e os requisitos de armazenamento de informação
- 4. Desenvolver os modos/estados operacionais baseado nos requisitos.
- 5. Avaliar as interfaces internas e externas do sistema de processamento, para estabelecer as arquiteturas e topologias candidatas
- 6. Selecionar uma arquitetura baseline
- 7. Criar uma especificação de arquitetura, modos, estados, para criar a documentação dos sistemas computacionais
- 8. Estabelecer uma abordagem de implementação e seus processos
- 9. Determinar a estratégia de entrega e de uso dos produtos

SRR

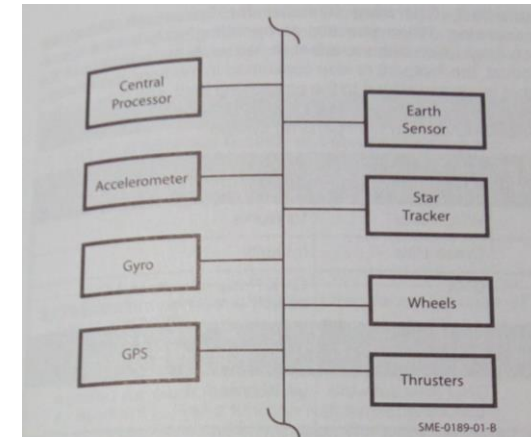
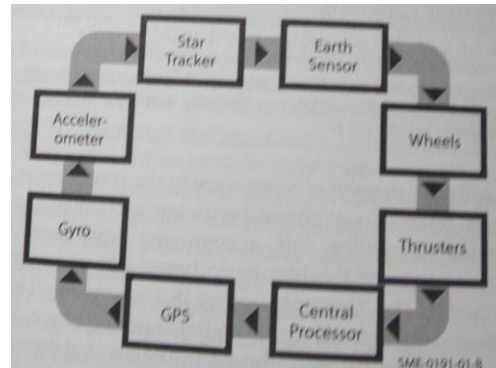
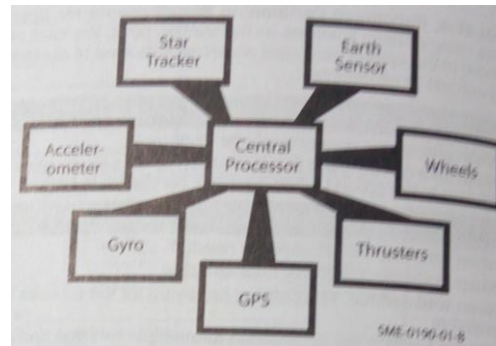
PDR

Questões gerais para levantar requisitos:

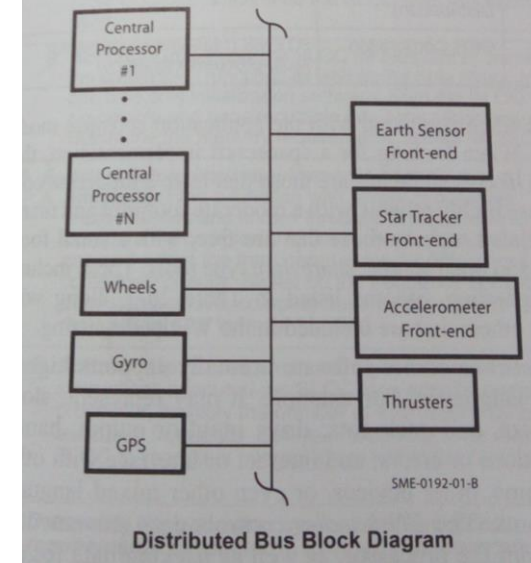
- O que o sistema deve fazer?
- Por que deve ser feito?
- Como podemos realizar e quais são as alternativas?
- Quais funções podem ser alocadas para cada parte de um sistema?
- As funções técnicas são possíveis?
- O sistema é testável?

Avaliação das Arquiteturas

- Topologia
- Estrutura dos dados
- Arquitetura do hardware
- Arquitetura do software



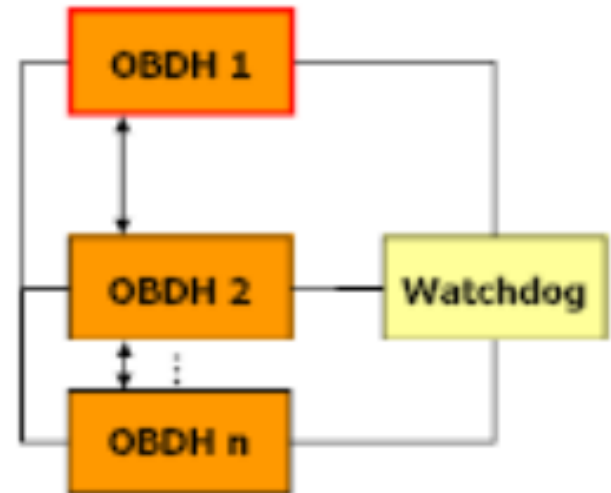
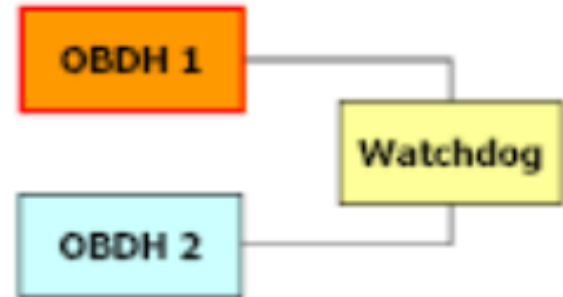
Federated Bus Block Diagram



Distributed Bus Block Diagram

Conceitos de Redundância

- Cold (Fria)
 - Apenas um computador é energizado e está ativo.
 - Troca por um terceiro elemento
- Hot (Quente)
 - Ambas estão energizadas e ativas
 - Troca pelos redundantes ou por um terceiro elemento.



Design de Software

Requisitos e
Objetivos da Missão

Computadores

1. Listar todas as funções
de softwares alocadas
num computador

2. Quebrar as funções
em unidades menores

- Quebrar até onde for necessário.
- Agrupar funções e utilidades em bibliotecas, ou em instancias únicas.

3. Definir a frequência de
execução (RT)

- Definir as funções que são críticas no tempo.

4. Estimar linhas de
código e memória

- Por similaridade com outras implementações, ou conhecimento prévio.
- Bottom-up

5. Estimar tempo de
processamento

- Fazer uma previsão de crescimento e maiores necessidades durante o voo.

6. Determinar o sistema
operacional.

- Identificar todo o código
- Complexidade do S.O. é relativo a quantidade de tarefas, interrupções, e I/O necessárias, bem como o computador escolhido.

7. Determinar margens
de crescimento

Particionamento das Funções Lógicas

Realiza Processamento no Espaço <ul style="list-style-type: none"> Quando timing é crítico. Quando dado deve ser processado para reduzir o pacote pro solo. 	Realiza Processamento em Solo <ul style="list-style-type: none"> Quando interação humana é necessária Quando o downlink é satisfatório
Realiza Processamento em Hardware <ul style="list-style-type: none"> Quando é necessário alta performance Quando o processamento é muito matemático Quando existe um hardware disponível/bem definido/custo acessível Quando a função deve ser replicada em múltiplos sats. 	Realiza Processamento em Software <ul style="list-style-type: none"> Quando o processamento é muito complexo Quando ocorre mudanças após o hardware ter sido comprado Quando o hardware é muito caro, e pode ser substituído por SW. Quando há muita memória inutilizada
Aloca processos entre B. de Serviços / Carga útil <ul style="list-style-type: none"> Quando é necessário ajustar o protocolo Quando o funcionamento da Carga é Crítico 	Não Aloca processos entre B. de Serviços / Carga Útil <ul style="list-style-type: none"> Quando o processamento na carga útil é mínimo Quando existe sinergia entre padrões
Aloca Processamento Através das Organizações <ul style="list-style-type: none"> Quando há impeditivos (contratos, geográficos, etc) para comunicação Inter organizacional. Quando há um controle muito rígido do destino dos dados 	Não Aloca Processamento Através das Organizações <ul style="list-style-type: none"> Quando o projeto é pequeno Quando o subsistema é tão completo que exige uma organização específica.

Softwares tem cinco atividades principais

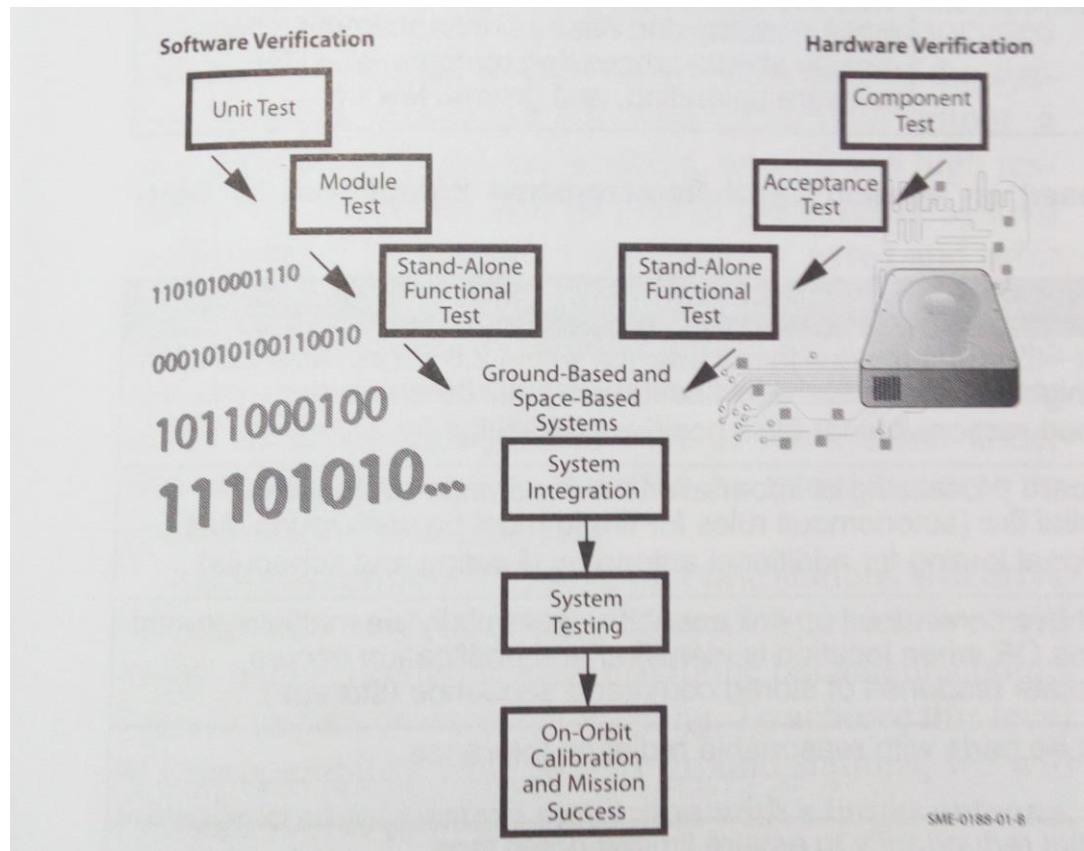
- Atividades do **sistema operacional** – controles de recursos, tarefas, interrupções, etc.
- Atividades de **gestão do sistema** – detecção e correção de erros, eventos programados,
- Atividades de **controle** – determinação e controle de órbita e atitude
- Atividades de **comando e gestão de dados** – coleta dados de status dos subsistemas.
- Atividades de **gestão das cargas úteis** – monitoramento, gerenciamento, e interfaces com a carga útil.

- Características “físicas” (peso / volume / consumo / capacidade / velocidade)
- Radiação
- Usa um firmware específico ou um sistema operacional
- Tipos/quantidades de Entradas e Saídas
- Tipos de memórias
- Disponibilidade

Desenvolvimento de Software

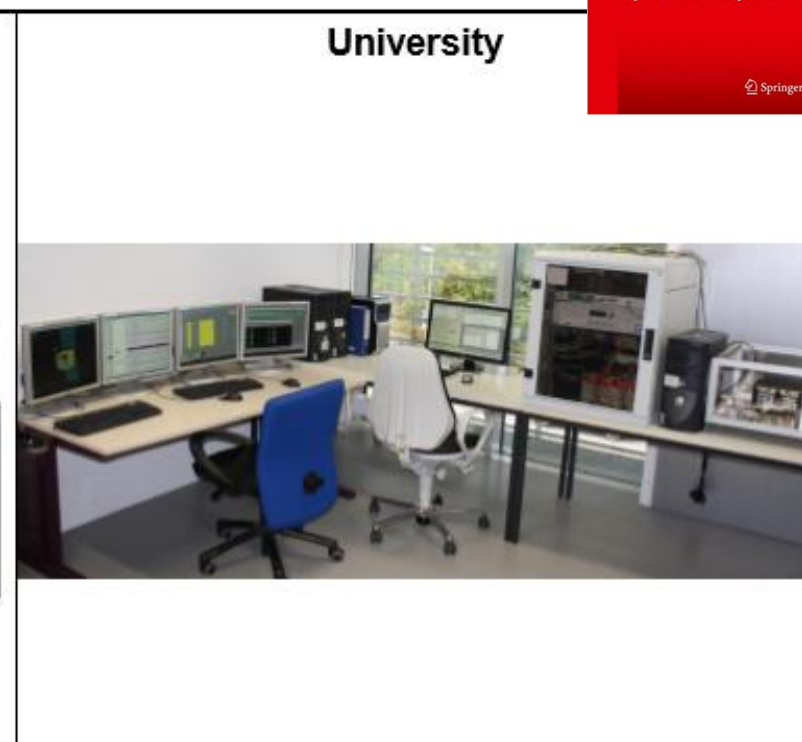
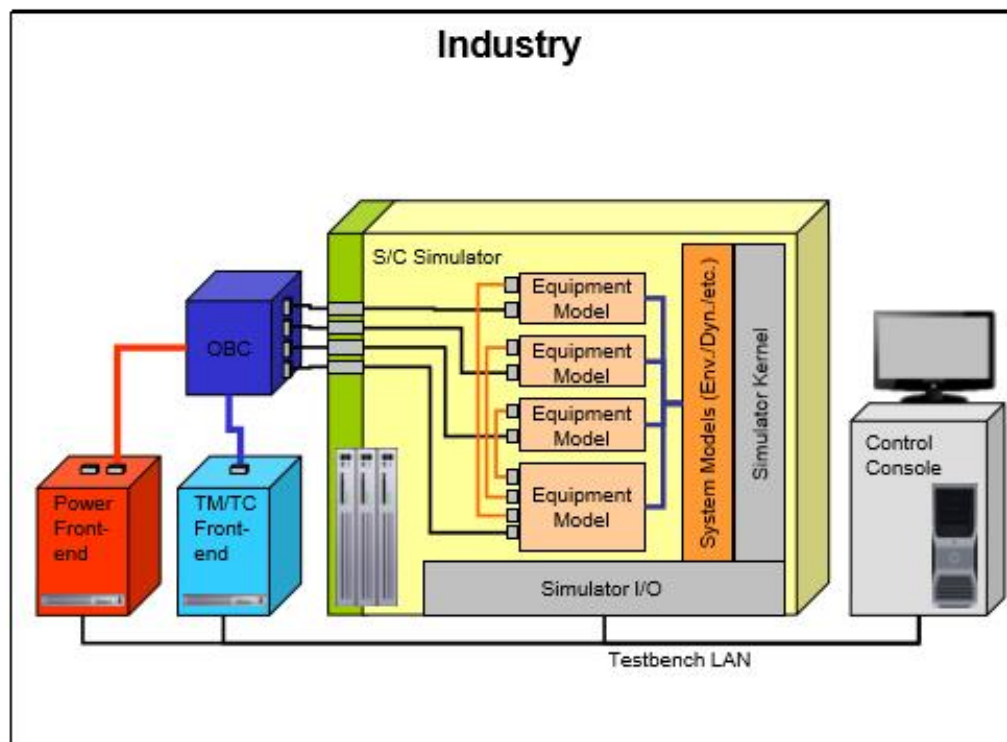
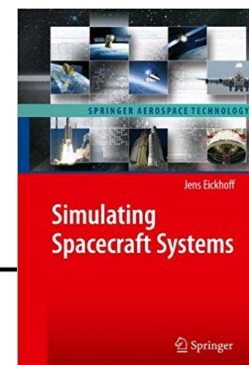
- Padrões (MIL-STDs, IEEE, SAE, ANSI, ISO)
- CMMI (Capability Maturity Model Integration)
- Disponibilidades e considerações
 - **Host machine** – ambiente de desenvolvimento
 - **Target machine** – computador onde o software vai ser embarcado
 - **Cross-compiler** – compila software para outra arquitetura
- Ferramentas de Modelagem
- Estilos/Orientações de Desenvolvimento

Testes e Integração



Simulation based functional Verification

Onboard Software testing on Satellite Testbench



© Eickhoff, J.: Simulating Spacecraft Systems, Springer, 2009



Sistemas Operacionais....

FreeRTOS™

The **Market Leading**, De-facto Standard and Cross Platform Real Time Operating System (RTOS). Don't Let Your RTOS Lock You In.

Immediate Free Download and Use ★ Feature Rich ★ Tiny Footprint ★ Easy To Use Pre-configured Projects ★ Can Be Used In Commercial Applications ★ Massive User Community ★ Free Support ★ Optional Commercial Licensing/Support ★ Strict Coding Standard ★ Safety Critical Version Available ★ Tickless Mode for Low Power Applications



Products ▾ Services Documentation Blog About Us Contact Us



An Accessible, Powerful Solution for any Mission

An open source, integrated platform built to run on any satellite subsystem. KubOS flight software focuses on managing your satellite bus so you can focus on building your mission. Packaged into an integrated distribution, KubOS is comprised of three parts: a customized operating system (RTOS or Linux), Kubos' hardware abstraction layer (HAL), and Kubos Core flight middleware. With an SDK built for developers, KubOS is the developer's choice.

Protocolos de Comunicação

- CAN
- I2C AD-HOC
- I2C CSP
- SERIAL AD-HOC
- Driven by Vendor