

Sistemas Operacionais Embarcados

Plano de Ensino

Plano de Ensino

- Descrição geral
- Formato das aulas
- Controle de presenças
- Critérios de avaliação
- Ementa
- Programa
- Bibliografia

Descrição geral

- Disciplina obrigatória ao curso de Engenharia Eletrônica/FGA, sobre o desenvolvimento de sistema embarcados utilizando sistemas operacionais.
- Pré-requisitos:
 - ELETRÔNICA EMBARCADA FGA0096 ou
 - MICROPROCESSADORES E MICROCONTROLADORES FGA0182 ou
 - FUNDAMENTOS DE SISTEMAS OPERACIONAIS FGA0170
- Professor:
 - Diogo Caetano Garcia / diogogarcia@unb.br / Eng. Eletrônica

Formato das aulas

- O curso consistirá em aulas síncronas via video-conferência na plataforma Microsoft Teams, nas segundas, quartas e sextas, sempre das 14:00 às 15:50.
- As aulas síncronas serão gravadas e disponibilizadas para acompanhamento assíncrono também na plataforma Microsoft Teams.
- Todo o conteúdo, listas de exercícios, provas e entrega de trabalhos estarão disponíveis via página no Moodle Aprender 3.
- O aluno pode tirar dúvidas via fórum de dúvidas no Moodle da disciplina, ou pode agendar por e-mail uma reunião via video-conferência.

Controle de presenças

- A presença dos alunos será conferida via lista de participantes nas video-conferências pela plataforma Microsoft Teams.
- **Favor conferir seu nome no Microsoft Teams, para evitar erros nas listas de presença.**
- Alunos que não tiverem condições de acompanhar as aulas sincronamente deverão assistir às aulas gravadas e indicar que o fizeram pela plataforma Teams.

Critérios de avaliação

- A avaliação na disciplina é dividida da seguinte forma:
 - Dois testes, cada um valendo 20% da nota final;
 - Projeto da disciplina, valendo 30% da nota final;
 - 4 pontos de controle do projeto da disciplina, cada um valendo 5% da nota final;
 - Respostas dos questionários das aulas no Moodle, valendo 10% da nota final.

Critérios de avaliação

Calendário_2021_1 : Sheet1

Julho							Agosto							Setembro							Outubro							Novembro						
D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S
				1	2	3	1	A7	3	A8	5	PC1	7				A18	2	PC2	4						1	2		1	2	3	4	5	6
4	5	6	7	8	9	10	8	A9	10	A10	12	A11	14	5	A19	7	A20	9	A21	11	3	A27	5	A28	7	PC3	9	7	8	9	10	11	12	13
11	12	13	14	15	16	17	15	A12	17	A13	19	T1	21	12	A22	14	A23	16	T2	18	10	A29	12	A30	14	A31	16	14	15	16	17	18	19	20
18	A1	20	A2	22	A3	24	22	A14	24	A15	26	A16	28	19	A24	21	A25	23	A26	25	17	A32	19	A33	21	PC4	23	21	22	23	24	25	26	27
25	A4	27	A5	29	A6	31	29	A17	31					26	27	28	29	30			24	A34	26	A35	28	PRJ	30	28	29	30				
																					31													
Aulas teóricas							Testes							Feriados							Férias													
Pontos de controle							Entrega do projeto							Semana Universitária																				

Sheet1

Critérios de avaliação

- Os alunos formarão duplas para montar um projeto final de maior dificuldade, cobrindo os tópicos vistos em sala de aula. Os projetos serão propostos pelos alunos, e será apresentado um protótipo em funcionamento e o relatório do mesmo, na forma de relatório científico com formatação IEEE, contendo os seguintes tópicos:
 - Resumo (1 ponto) : Apresentar o projeto de forma geral e em poucas palavras.
 - Introdução (1 ponto) : Descrição do problema a ser resolvido, revisão bibliográfica do estado-da-arte do assunto.
 - Desenvolvimento (5 pontos) : Solução, com justificativa baseada em fundamentos teóricos. O desenvolvimento deverá conter:
 - Descrição do hardware (2 pontos), incluindo a BOM (bill of materials) e a montagem (diagrama de blocos, esquemático, descrição textual etc.).
 - Descrição do software (3 pontos), apresentando e justificando o algoritmo desenvolvido. Esta seção do relatório NÃO CONSISTE em simplesmente replicar o código, e sim em explicar como ele funciona, com base em fluxogramas, pseudo-códigos etc. O código deverá ser apresentado como um apêndice do relatório.
 - Resultados (2 pontos): Explicar os experimentos definidos para validar o projeto proposto, seguido de análise crítica dos resultados esperados e obtidos. Em caso de divergências, apontar as possíveis causas.
 - Conclusão (1 ponto): Retomar sucintamente os principais pontos do relatório: descrição do problema, solução utilizada e resultados obtidos.

Cr terios de avalia  o

- Os pontos de controle ser o pr vias do projeto final. Cada dia de atraso na apresenta  o dos pontos de controle acarreta na perda de 0,5 ponto da nota do mesmo. Ser o esperados os seguintes resultados para cada ponto de controle:
 - PC1: proposta do projeto (justificativa, objetivos, requisitos, benef cios, revis o bibliogr fica).
 - PC2: prot tipo funcional do projeto, utilizando as ferramentas mais b sicas da placa de desenvolvimento, bibliotecas prontas etc.
 - PC3: refinamento do prot tipo, acrescentando recursos b sicos de sistema (m ltiplos processos e threads, pipes, sinais, sem foros, MUTEX etc.).
 - PC4: refinamento do prot tipo, acrescentando recursos de Linux em tempo real.

Critérios de avaliação

- Todos os relatórios dos pontos de controle serão entregues via Moodle. Em caso de cópias (integrais ou parciais), os relatórios ficarão com nota ZERO. Os diferentes templates com formatação IEEE estão disponíveis em:
 - [Word](#)
 - [Unix LaTeX](#)
 - [Windows LaTeX](#)

Ementa

- Introdução aos sistemas embarcados
- Introdução aos sistemas operacionais Linux
- Recursos do sistema operacional
- Desenvolvimento para sistemas embarcados
- Subsistema de I/Os.
- Introdução ao RTOS
- Introdução aos LKM e aos device drivers

Programa

- Introdução aos sistemas embarcados
- Introdução aos sistemas operacionais Linux
 - Obtendo informações do sistema
 - Principais comandos
 - Estrutura de diretórios
 - Instalação padrão de programas

Programa

- Recursos do sistema operacional
 - Processos, sinais e threads
 - Comunicação e sincronismo entre processos
 - Programação para redes (sockets)
 - Tarefas
 - Alarmes
 - MUTEX
 - Semáforos
 - Variáveis condicionais
 - Fila de mensagens

Programa

- Desenvolvimento para sistemas embarcados
 - Conceito de cross-platform
 - Componentes e suas funções (host, target, interface de comunicação, etc)
 - Processo de geração de imagens
 - Utilização de makefiles
 - Inicialização em Sistemas Embarcados

Programa

- Desenvolvimento para sistemas embarcados
 - Ferramentas para o Target System
 - Transferência de imagens
 - Cenários de boot do target
 - Seqüência de inicialização do target

Programa

- Subsistema de I/Os.
 - Conceito básico de I/Os
 - O subsistema de I/Os
 - Modo de transferência serial e em blocos
 - Funções de um subsistema de I/O
- Introdução ao RTOS
 - Características dos sistemas RT
 - Conceitos de Hard Real-Time e Soft Real-Time
 - Conceito de latência e Jitter
 - Introdução ao framework Xenomai

Programa

- Introdução aos LKM e aos device drivers
 - Desenvolvimento de um módulo do kernel
 - Comandos do modutils
 - Conceitos básicos sobre device driver
 - Estrutura de um device driver
 - Funções de um device driver
 - Instalação e carregamento de um device driver

Bibliografia

Básica

1. Doug Abbott, Embedded Technology: Linux for Embedded and Real-Time Applications, 2nd Ed., Newnes, USA, 2006.
2. Mark Mitchell, Jeffrey Oldham e Alex Samuel, Advanced Linux Programming, New Riders, USA, 2001.
3. Christopher Hallinan, Embedded Linux Primer: A Practical Real-World Approach, Prentice Hall, USA, 2006.
4. Qing Li e Caroline Yao, Real-Time Concepts for Embedded Systems, CMP, USA, 2003.

Bibliografia

Complementar

5. Rajaram Regupathy, Bootstrap Yourself with Linux-USB Stack : Design, Develop, Debug, and Validate Embedded USB, Course Technology, USA, 2011.
6. Daniel Bovet e Marco Cesati, Understanding the Linux Kernel, 3rd Ed., O'Reilly, USA, 2005.
7. Jonathan Corbet, Alessandro Rubini e Greg Kroah-Hartman, Linux Device Drivers, 3rd Ed., O'Reilly, USA, 2005.
8. Karim Yaghmour, Jon Masters e Gilad Ben-Yossef, Building Embedded Linux Systems, 3rd Ed., O'Reilly, USA, 2003.