

# 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](mailto:diogogarcia@unb.br) / Eng. Eletrônica

# Formato das aulas

- O curso consistirá em aulas teóricas presenciais, nas segundas, quartas e sextas, sempre das 14:00 às 15:50.
- 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 a cada aula, sendo necessário assistir a pelo menos 75% das aulas para não receber menção SR ao final do semestre.

# 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

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

# 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.



# 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:
  - 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.

# 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:
  - 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
    - Apresenta  o do tema escolhido pela dupla
    - Defini  o do problema a ser resolvido
    - Pesquisa realizada: produtos j  existentes, artigos etc.
    - **Texto:** introdu  o, incluindo justificativa, objetivos, requisitos, benef cios e revis o bibliogr fica.

# Critérios de avaliação

- Os pontos de controle serão prévias do projeto final. Cada dia de atraso na apresentação dos pontos de controle acarreta na perda de 0,5 ponto da nota do mesmo. Serão esperados os seguintes resultados para cada ponto de controle:
  - PC2: funcionamento básico de cada parte fundamental:
    - Conectar partes fundamentais ao Raspberry Pi, de acordo com o projeto proposto (por exemplo, GPS, temperatura, envio de e-mail, câmera digital, reconhecimento facial etc.)
    - Demonstrar que é possível utilizar cada uma das partes com o Raspberry Pi;
    - Permitido utilizar linguagens interpretadas (Python, Octave etc.) e programas prontos.
    - **Texto:** acrescentar descrição de *hardware*, descrição de *software* e resultados obtidos.

# Cr terios de avalia  o

- Os pontos de controle s 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. S o esperados os seguintes resultados para cada ponto de controle:
  - PC3: refinamento do prot tipo
    - Conectar as partes fundamentais.
    - Acrescentar recursos b sicos de sistema (m ltiplos processos e threads, pipes, sinais, sem foros, MUTEX etc.).
    - Desenvolver em linguagem C e/ou C++.
    - **Texto:** atualizar descri  o de *hardware*, descri  o de *software* e resultados obtidos.

# Cr terios de avalia  o

- Os pontos de controle s 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. S o esperados os seguintes resultados para cada ponto de controle:
  - PC4: refinamento do prot tipo
    - Conectar as partes fundamentais.
    - Acrescentar recursos b sicos de sistema (recursos de Linux em tempo real etc.).
    - Desenvolver em linguagem C e/ou C++.
    - **Texto:** atualizar descri  o de *hardware*, descri  o de *software* e resultados obtidos.

# Critérios de avaliação

- Uma analogia com o projeto: montagem de um quebra-cabeças
  - PC1: escolha do quebra-cabeças
  - PC2: disposição de **todas as peças** do quebra-cabeças sobre a mesa
  - PC3: montagem do quebra-cabeças
  - PC4: montagem do quebra-cabeças

# 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.