

PLANO DE ENSINO

ESCOLA: *Politécnica*

DISCIPLINA: Programação de Software Básico (*turma 30*)

CÓDIGO: 46526 **CRÉDITOS:** 02

SEMESTRE: 2025/1

CARGA HORÁRIA: 30 h

(X) Presencial

() On-line

() Extensionista

CARGA HORÁRIA DE TRABALHO DISCENTE EFETIVO: 3 h

PROFESSOR

NOME

Eduarda Rodrigues Monteiro

CORREIO ELETRÔNICO

eduarda.monteiro@pucrs.br

EMENTA:

Estudo de linguagem de programação adequada para os chamados sistemas “de software básico” ou que exijam alto desempenho. Aprendizado de uma linguagem de programação que tenha acesso direto a recursos de hardware e sistema operacional, e que propicie implementações eficientes.

OBJETIVOS DE APRENDIZAGEM/COMPETÊNCIAS:

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

- 1. Familiarizar-se com uma linguagem tipicamente usada para desenvolvimento de sistemas de software básico e de alto desempenho;*
- 2. Aprender detalhes dos aspectos da linguagem tipicamente envolvidos na capacidade de acesso e manipulação dos recursos de baixo nível da máquina;*
- 3. Exercitar a capacidade de implementação de programas de software básico;*
- 4. Continuar os estudos em programação avançada.*

CONTEÚDOS:

UNIDADE 01 - A linguagem C de programação

1.1. A função main e estrutura básica de um programa

1.2. Tipos básicos

1.3. Entrada e saída: scanf, printf

1.4. Expressões

1.4.1. Operadores aritméticos, relacionais e lógicos

1.4.2. Operador de atribuição

1.4.3. Operadores bitwise e manipulação de bits

1.4.4. Coerção (cast)

1.5. Estruturas de controle de fluxo: if, else, switch, for, while, break, continue

1.6. Tipos estruturados

1.6.1. Arrays

1.6.2. Representação de strings como array de char null-terminated

1.6.3. Structs

1.6.4. Unions

1.6.5. Arquivos

1.7. Funções

1.7.1. Declarações, definições, protótipos

1.7.2. Passagem de parâmetros: por valor, por referência

1.8. Organização de um programa

1.8.1. Declarações, definições, protótipos

1.8.2. Diretivas de pré-processamento

1.8.3. Bibliotecas e includes; arquivos .h e .c

1.8.4. Compilação em separado, makefiles

1.8.5. Criação de bibliotecas estáticas e dinâmicas (DLLs e shared objects)

UNIDADE 02 - Ponteiros

- 2.1. Ponteiros e endereços de memória
- 2.2. Declaração e uso de ponteiros; operadores de endereço (&) e dereferência (*)
- 2.3. Passagem de parâmetros usando ponteiros (simulação da passagem por referência de modo "não-transparente")
- 2.4. Dualidade entre ponteiros e vetores; aritmética dos ponteiros; ponteiros para "void"
- 2.5. Alocação dinâmica de memória, comparação entre stack e heap
- 2.6. Ponteiros para structs e operador ->
- 2.7. Atributos (campos) do tipo ponteiro
- 2.8. Ponteiros para funções
- 2.9. Segmentation fault e memory leak

PROCEDIMENTOS METODOLÓGICOS:

*As atividades são realizadas em sala de aula e em laboratório, de acordo com o cronograma.
As aulas são expositivas e dialogadas, com uso de exemplos e resolução de exercícios.*

Carga horária total de atividades extraclasse: 3 horas.

A carga horária está distribuída entre os trabalhos.

Atividade - Carga horária estimada

T1 – 1,5h

T2 – 1,5h

AVALIAÇÃO:

As provas são individuais, sem consulta. Trabalhos e exercícios podem ser realizados em grupo, conforme constar no enunciado divulgado.

O cálculo do G1 é realizado segundo a fórmula abaixo:

$$G1 = \frac{(2 * P1 + T1 + T2)}{4}$$

Onde:

P1 – avaliação escrita e individual, abrangendo as unidades I e II, conforme cronograma.

T1 e T2 são trabalhos práticos, conforme cronograma.

A prova PS substitui a nota de P1, somente em caso de ausência.

O grau G2 é calculada conforme previsto no Regimento.

BIBLIOGRAFIA:

BÁSICA

1. DAMAS, Luís M. D.. *Linguagem C*. 10 ed. Rio de Janeiro: LTC, c2007.
2. DEITEL, Harvey M; DEITEL, PAUL. *C: como programar*. 6. ed. Porto Alegre: Pearson Education, 2011.
3. PRATA, STEPHEN. *C Primer Plus*. 6a ed. Addison-Wesley Professional, 2013. 1080p.

COMPLEMENTAR

1. KING, K. N. *C Programming: A Modern Approach*. 2a. ed. W. W. Norton & Company, 2008. 832p.
2. KLEMENS, BEN. *21st Century C: C Tips from the New School*. 2a. ed. O'Reilly Media, 2014. 410p.
3. PETERSEN, Richard. *Linux: the complete reference*. Berkeley: Osborne McGraw-Hill, 1996. 838 p.
4. VAN DER LINDEN, P. *Expert C Programming: Deep C Secrets*. Mountain View: SunSoft, 1994. 353 p.
5. LECKY-THOMPSON, GUY W. *Just Enough C/C++ Programming*. Boston, MA, USA: Course Technology Cengage Learning, 2007.

CRONOGRAMA DE AULAS:

O cronograma pode ser alterado ao longo do semestre. O cronograma atualizado deve ser consultado via Sarc ou Moodle.