

Plano de Aula**Curso:** ENGENHARIA DE COMPUTAÇÃO**Disciplina:** SISTEMAS HARDWARE-SOFTWARE**Turma:** SISHARD_SENGCOMA**Curriculo:** 202361**Período Letivo:** 2026 / 61**Professores:** Fabio Aparecido Gamarra Lubacheski**E-mails:** fabioagl@INSPER.EDU.BR**Horário das aulas:** Seg 16:30 até 18:30 e Qui 16:30 até 18:30**Programa**

Data	Questão / Problema / Desafio	Fundamentos / Conteúdo	Evidências de Aprendizado	Programação / Atividades
Aula 1 09/02/2026	Mutirão C	Programação em C	Prova do Mutirão C	Atividades práticas de introdução a programação em C. Integrada com as disciplinas Sistemas Embaçados e Desafios de Programação.
Aula 2 12/02/2026	Mutirão C	Programação em C	Prova do Mutirão C	Atividades práticas de introdução a programação em C. Integrada com as disciplinas Sistemas Embaçados e Desafios de Programação.
Aula 3 19/02/2026	Como a CPU representa dados em baixo nível?	Representação de inteiros na CPU: números binários e hexadecimais, complemento de dois, operações bit a bit. Leitura: Cap 2 CS:APP; Cap 9 Stallings	Avaliação Intermediária; Atividade para entrega	Exercícios sobre conversão de bases numéricas
Aula 4 23/02/2026	Como a CPU representa dados em baixo nível?	Representação de ponteiros, arrays e estruturas em RAM. Leitura: Cap 2 CS:APP; Cap 9 Stallings	Avaliação Intermediária; Atividade para entrega	Atividades práticas usando gdb para analisar estruturas em memória.
Aula 5 26/02/2026	Como a CPU representa e executa um programa escrito em C?	Arquitetura de computadores: chamadas de função e pilha Assembly: Exercícios com chamadas de função e operações com inteiros Leitura: Cap 3.5 CS:APP	HackerLab; Avaliação Intermediária; Atividade para entrega	Atividades práticas no papel e no computador usando gdb
Aula 6 02/03/2026	Como a CPU representa e executa um programa escrito em C?	Arquitetura de computadores: registradores, memória e endereçamento Assembly: Organização de um executável no disco e na memória Leitura: Cap 3.1 – 3.4 CS:APP	HackerLab; Avaliação Intermediária; Atividade para entrega	Atividades práticas no papel.
Aula 7 05/03/2026	Como a CPU representa e executa um programa escrito em C?	Assembly: expressões booleanas e estruturas de controle condicionais; e tradução do Assembly em C (gotoC).	HackerLab; Avaliação Intermediária; Atividade para entrega	Atividades práticas no papel e no computador usando gdb
Aula 8 09/03/2026	Como a CPU representa e executa um programa escrito em C?	Assembly: condicionais e conversões de funções em C	HackerLab; Avaliação Intermediária; Atividade para entrega	Atividades práticas no papel e no computador usando gdb
Aula 9 12/03/2026	Como a CPU representa e executa um programa escrito em C?	Assembly: estruturas de controle de loop Leitura: Cap 3.6 CS:APP	HackerLab; Avaliação Intermediária; Atividade para entrega	Atividades práticas no papel e no computador usando gdb
Aula 10 16/03/2026	Como a CPU representa e executa um programa escrito em C?	Revisão; organização de executável no disco e na memória. Assembly: variáveis locais, strings Leitura: Cap 3.7 – 3.8 CS:APP	HackerLab; Avaliação Intermediária; Atividade para entrega	Atividades práticas no papel e no computador usando gdb
Aula 11 19/03/2026	Como a CPU representa e executa um programa escrito em C?	Revisão; organização de executável no disco e na memória Assembly: arrays Leitura: Cap 3.7 – 3.8 CS:APP	HackerLab; Avaliação Intermediária; Atividade para entrega	Atividades práticas no papel e no computador usando gdb
Aula 12 23/03/2026	Revisão para AI	Exercícios práticos de revisão - Simulado da AI	Exercícios práticos de revisão	Atividades práticas no papel e no computador
Aula 13 26/03/2026	AVALIAÇÃO INTERMEDIÁRIA	AVALIAÇÃO INTERMEDIÁRIA	AVALIAÇÃO INTERMEDIÁRIA	AVALIAÇÃO INTERMEDIÁRIA

Data	Questão / Problema / Desafio	Fundamentos / Conteúdo	Evidências de Aprendizado	Programação / Atividades
Aula 14 30/03/2026	AVALIAÇÃO INTERMEDIÁRIA	AVALIAÇÃO INTERMEDIÁRIA	AVALIAÇÃO INTERMEDIÁRIA	AVALIAÇÃO INTERMEDIÁRIA
Aula 15 06/04/2026	E se meu programa precisar de mais memória?	A função malloc e a alocação dinâmica de estruturas, arrays e strings. Uso da ferramenta valgrind para checagem de memória Leituras: Cap 16 Kochan (Prog in C); Cap 9.9 CS:APP	Avaliação Final; Atividade para entrega	Atividades práticas para entrega usando valgrind para checagem de memória.
Aula 16 09/04/2026	E se meu programa precisar de mais memória?	Tipos Abstratos de dados e alocação dinâmica de memória Leituras: Cap 6 K&R	Avaliação Final; Atividade para entrega	Atividades práticas para entrega usando valgrind para checagem de memória.
Aula 17 13/04/2026	O que é um sistema operacional? Como os programas se comunicam com ele?	Sistemas operacionais: processos. Explorando chamadas de sistemas para criação de processo (fork) e espera de seus resultados (wait). Leitura: Cap 8 CS:APP, Leitura complementar sobre escalonamento de processos: Cap 5 Silberschatz	Avaliação Final; Atividade para entrega	Atividade prática com desenvolvimento de programas em sala de aula. É dada ênfase na utilização do man para consulta da documentação das chamadas POSIX
Aula 18 16/04/2026	O que é um sistema operacional? Como os programas se comunicam com ele?	Sistemas operacionais: processos. Explorando chamadas de sistemas para execução de programas (exec) Leitura: Cap 8 CS:APP,	Avaliação Final; Atividade para entrega	Atividade prática com desenvolvimento de programas em sala de aula. É dada ênfase na utilização do man para consulta da documentação das chamadas POSIX
Aula 19 23/04/2026	O que é um sistema operacional? Como os programas se comunicam com ele?	Sistemas operacionais: processos. Explorando chamadas de sistemas para entrada/saída Leitura: Cap 8 CS:APP,	Lab de Processos; Avaliação Final; Atividade para entrega	Atividade prática com desenvolvimento de programas em sala de aula. É dada ênfase na utilização do man para consulta da documentação das chamadas POSIX
Aula 20 27/04/2026	E se o SO quiser avisar um processo que algo ocorreu?	Sinais: conceitos, captura e bloqueio. Introdução à ideia de concorrência	Lab de Processos; Avaliação Final; Atividade para entrega	Atividade prática com desenvolvimento de programas em sala de aula. É dada ênfase na utilização do man para consulta da documentação das chamadas POSIX
Aula 21 30/04/2026	E se o SO quiser avisar um processo que algo ocorreu?	Captura e bloqueio de sinais: teoria e exercícios práticos	Lab de Processos; Avaliação Final; Atividade para entrega	Atividade prática com desenvolvimento de programas em sala de aula. É dada ênfase na utilização do man para consulta da documentação das chamadas POSIX
Aula 22 04/05/2026	É possível dividir o trabalho em partes que executem ao mesmo tempo?	Introdução ao conceito de threads e sua utilização Criação de threads usando pthreads. Leitura: Cap 12 CS:APP	Lab de Processos; Avaliação Final; Atividade para entrega	Atividade prática com desenvolvimento de programas em sala de aula.
Aula 23 07/05/2026	É possível dividir o trabalho em partes que executem ao mesmo tempo?	Tarefas IO- e CPU-bound. Sincronização usando Mutex	Lab de Processos; Avaliação Final; Atividade para entrega	Atividades de simulação de problemas de concorrência no papel e sincronização usando Mutex
Aula 24 11/05/2026	É possível dividir o trabalho em partes que executem ao mesmo tempo?	Tarefas IO- e CPU-bound. Sincronização usando semáforos. Leitura: Cap 12 CS:APP	Lab de Processos; Avaliação Final; Atividade para entrega	Atividades de simulação de problemas de concorrência no papel e sincronização usando semáforos
Aula 25 14/05/2026	É possível dividir o trabalho em partes que executem ao mesmo tempo?	Aula estúdio para projeto (Lab de Processos)	Lab de Processos; Avaliação Final; Atividade para entrega	Aula estúdio para projeto (Lab de Processos)
Aula 26 18/05/2026	É possível dividir o trabalho em partes que executem ao mesmo tempo?	Tarefas IO- e CPU-bound. Sincronização usando semáforos. Leitura: Cap 12 CS:APP	Avaliação Final; Atividade para entrega	Atividades de simulação de problemas de concorrência no papel e sincronização usando semáforos
Aula 27 21/05/2026	É possível dividir o trabalho em partes que executem ao mesmo tempo?	Problemas clássicos em programação multi-thread: Produtor-consumidor, Leitores escritores. Leitura: Cap 12 CS:APP	Avaliação Final; Atividade para entrega	Exercícios práticos no papel sobre sincronização usando semáforos
Aula 28 25/05/2026	Revisão para AF	Exercícios práticos de revisão - Simulado da AF	Exercícios práticos de revisão	Atividades práticas no papel e no computador
Aula 29 28/05/2026	AVALIAÇÃO FINAL	AVALIAÇÃO FINAL	AVALIAÇÃO FINAL	AVALIAÇÃO FINAL
Aula 30 01/06/2026	AVALIAÇÃO FINAL	AVALIAÇÃO FINAL	AVALIAÇÃO FINAL	AVALIAÇÃO FINAL

Atividades Práticas Supervisionadas (APS)

Atividade

Observação

Critério de Avaliação

Nome da Avaliação	Sigla	Peso em %
Avaliação Intermediária	AI	20
Avaliação Final	AF	25
Laboratórios	Lab	40
Prova Mutirão	MC	5
Atividades	Atv	10

Orientações gerais:

Critério de avaliação na disciplina

$$NS = 0,10*Atv + 0,20*AI + 0,30*AF + 0,40*Lab$$

$$NC = 0,10*Atv + 0,20*AI + 0,25*AF + 0,35*Lab + 0,05*MC$$

Para aprovação as CONDIÇÕES abaixo devem ser atendidas:

$$((AI + AF) / 2) \geq 4,0 \text{ E}$$

$$AI \text{ e } AF \geq 3,5 \text{ E}$$

$$Lab \geq 5,0$$

A média final (MF) é calculada com a seguinte fórmula

Se atendida as CONDIÇÕES: $MF = \max(NS, NC)$

Se NÃO atendida as CONDIÇÕES: $MF = \min(Atv, AI, AF, Lab, MC)$