



PCS3616

Programação de Sistemas

(Sistemas de Programação)

Visão Geral

Escola Politécnica da Universidade de São Paulo

Objetivos

- Familiarização com o funcionamento global de software de sistema computacional
- Desenvolver experimentos em computador
 - Ambiente (Expressões Regulares, scripts)
 - Máquinas/Algoritmos (Máquina de Turing)
 - Software de Sistema (Montadores, Editores de Ligação. Relocadores)
 - Noções de Compiladores (Front-end)
- Ferramenta para automatização de geração de código, otimização em Compiladores (Back-end).

Programa

Semana	Aula	Assunto
09-11 jan	1	Instalação do ambiente: Linux, scripts
16-18 jan	2	Ambiente: Linux, programas básicos, Expr. Reg.
01-03 fev	3	Exercícios usando MT
08-10 fev	4	MVN, programas em linguagem de máquina da MVN; A simulação da MVN (programar); Virtualização
15-17 fev	5	MVN, programas em linguagem de máquina da MVN; A simulação da MVN (programar); Virtualização
22-24 fev	6	Programas para a MVN em linguagem de montagem
06-08 mar	7	Montadores e ligadores para a MVN
13-15 mar	8	Alocador e carregador para a MVN
20-22 mar	9	Implementação de um monitor batch para a MVN
27-29 mar	10	Compilação: LEX, YACC (Xtext), etc.
03-05 abr	11	Compilação: LEX, YACC (Xtext), etc.
10-12 abr	12	Compilação back-end: LLVM
17-19 abr	13	Encerramento

Organização

- **Prática** \approx 100min
 - Exercícios individuais
 - Entregues no final da aula

Organização

- Material no Moodle do Stoa
 - <https://disciplinas.stoa.usp.br/course/view.php?id=31356>
- Professor
 - Ricardo Luis de Azevedo da Rocha
- Monitor
 - Não temos ainda

Bibliografia

■ Básica

- SAVAGE, J. “Models of Computation: Exploring the Power of Computing”,
- BRYANT R. E. and O’HALLARON, D. R. “Computer Systems: A Programmer’s Perspective”, 2010
- DONOVAN, J. “Systems Programming”, 1972
- RICARTE, I. “Introdução à Compilação”, 2008
- JOSÉ NETO, J. “Introdução à Compilação”, 1987
- LLVM – llvm.org; www.aosabook.org/en/llvm.html

■ Complementar

- LUCCHESI, C. L. “Aspectos Teóricos da Computação”, 1979
- SIPSER, Michael “Introdução à teoria da computação”, Thomson, 2007.
- BECK, L. L. “An introduction to Systems Programming”, Addison Wesley, 1997
- LOPES, B. C. and AULER, R. “Getting started with LLVM Core Libraries”, 2014
- PANDEY, M. and SARDA, S. “LLVM Cookbook”, 2015

Desenvolvimento de Software

- Desenvolver software não envolve só uma linguagem de programação
 - Métodos, Arcabouços (*frameworks*), Bibliotecas, Ferramentas, etc.
- Um aspecto importante é o *paradigma de programação*

“Forma de conceituar o que significa realizar computação e como tarefas executadas no computador devem ser estruturadas e organizadas.” (Budd, 2001)

BUDD, T. *An Introduction to Object-Oriented Programming*. 3rd Edition. Addison-Wesley. 2001. Cap. 1.

- A solução de um problema computacional é influenciada pelo paradigma seguido
 - Facilidade / dificuldade de representação

Paradigma de Programação

- Alguns paradigmas
 - **Imperativo:** estado global e comandos de mudanças de estado
 - *Linguagens:* Pascal, C e Cobol
 - **Funcional:** funções matemáticas (não afetam o estado)
 - *Linguagens:* Lisp, Haskell, ML e Scala
 - **Lógico:** lógica formal (ex.: lógica de 1ª ordem)
 - *Linguagens:* Prolog e Datalog
 - **Orientação a objetos:** abstração do mundo em objetos
 - *Linguagens:* C++ (*), Java, C#, Objective C, Smalltalk, Python(*)
- Algumas linguagens são *multiparadigma*, como as marcadas com (*)