
Sistemas de Operação

Apontamentos das Aulas Teóricas

Acetatos baseados nas aulas do Prof. Fernando Silva

Inês Dutra (teóricas) e Ricardo Rocha (práticas)
ines@dcc.fc.up.pt (gab. 1.31), ricroc@dcc.fc.up.pt (gab. 1.26)
<http://www.dcc.fc.up.pt/ines/aulas/0910/SO/SO.html>

Departamento de Ciência de Computadores

Faculdade de Ciências da Universidade do Porto

2009/2010

Sistemas de Operação

- Objetivos: estudar os conceitos, estrutura, e mecanismos dos sistemas de operação.
- Com que finalidade?
 - entender como um SO funciona para melhor aproveitar suas funcionalidades durante a programação de aplicações;
 - ter uma visão crítica das limitações de um SO;
 - ter a competência para propor e implementar melhoramentos.

Sistemas de Operação – Programa

● Aulas teóricas:

- Conceitos básicos de SO: sistemas batch, buffering e spooling, multiprogramação, multi-tarefa (*time-sharing*), sistemas distribuídos e sistemas tempo-real
- Gestão de Processos e Threads: criação, escalonamento para execução, temporização, sincronização, zonas críticas, comunicação
- Gestão de Memória: mono- e multiprogramação, partições físicas, swapping, paginação, segmentação, memória virtual
- Sistema de Ficheiros: representação de ficheiros em disco, gestão do espaço em disco, mecanismos de proteção
- Periféricos de Entrada/Saída
- Impasses (deadlocks) e segurança

Sistemas de Operação – Programa

- Aulas práticas:
 - Programação Avançada em Unix e em C:
 - implementação de pequenos programas que utilizam funcionalidades do SO
 - implementação de partes básicas de um SO

Bibliografia

- Andrew S. Tanenbaum, [Modern Operating Systems](#), 3rd edition, Prentice Hall, 2008.
- Silberschatz & Galvin & Gagne, [Operating System Concepts](#), 8th edition, John Wiley & Sons, 2009.
- W. Stallings, [Operating Systems: Internals and Design Principles](#), 6th edition, Prentice-Hall, 2009. (material complementar em <http://williamstallings.com/OS/OS6e.html>)
- R. Stevens & S. Rago, [Advanced Programming in the UNIX\(R\) Environment](#), 2nd edition, Addison-Wesley, 2005.
- Marc J. Rochkind, [Advanced UNIX Programming](#), 2nd edition, Addison-Wesley, 2004
- Mark Mitchell & Alex Samuel, [Advanced Linux Programming](#), New Riders, 2001.
- Warren W. Gay, [Advanced UNIX Programming](#), SAMS, 2000.

Método de Avaliação

- Aulas práticas e teóricas:
 - não há marcação de faltas, porém alunos que frequentam aulas teóricas e práticas têm maior chance de sucesso!
- Trabalhos práticos: TP
 - a realizar em grupos de 2 alunos
 - 2 trabalhos **NÃO** obrigatórios
- Exame escrito: EX (com componente prática)
 - valorização de 20 valores
- Testes: T1 e T2 (com componente prática)
 - valorização de 20 valores

Método de Avaliação

- Classificação final (CF):

$$CF = (T1 + T2 + EX)/num_aval$$

onde:

- num_aval = 1 (EX)
 - num_aval = 2 (EX e um de T1 ou T2)
 - num_aval = 3 (EX, T1 e T2)
- Nota importante: Copiar é crime!!!!