FACULDADE: CENTRO UNIVERSITÁRIO DE BRASÍLIA – UniCEUB
CURSO: CIÊNCIA DA COMPUTAÇÃO
DISCIPLINA: SISTEMAS OPERACIONAIS
CARGA HORÁRIA: 75 H. A. **ANO/SEMESTRE**: 2016/02
PROFESSOR: EDUARDO FERREIRA DOS SANTOS
HORÁRIOS: Terça às 07h20 e Sexta às 09h40

LISTA DE EXERCÍCIOS 01

EXERCÍCIO 01

1. Como funciona a interface homem máquina em um sistema operacional?
2. O que são sistemas operacionais de kernel monolítico? Descreva a interação entre os programas hospedados pelo sistema operacional e o hardware para esse tipo de SO.
3. O que são sistemas operacionais baseados em camadas? Descreva a interação entre os programas hospedados pelo sistema operacional e o hardware para esse tipo de SO.
4. O que são sistemas operacionais baseados em microkernel? Descreva a interação entre os programas hospedados pelo sistema operacional e o hardware para esse tipo de SO.
5. Para cada um desses tipos de SO, apresente um caso de uso onde a arquitetura do kernel favorece a sua utilização.

EXERCÍCIO 02

6. Explique a diferença entre os seguintes conceitos:
   (a) Tarefas;
   (b) Processos;
   (c) Threads.
7. Apresente um exemplo prático para ilustrar as diferenças que você definiu.
8. O que é a abordagem de Von Neumann? Como ela afeta a distribuição de tarefas em um Sistema Operacional?
9. Quais são os possíveis estados dos processos em um sistema operacional?
10. Qual é a diferença entre programação sequencial e multiprogramação? Apresente um exemplo para ilustrar suas justificativas.
11. Descreva como é a utilização do processador em cada um dos exemplos, descrevendo os possíveis estados dos processos em cada etapa.
12. Descreva, com exemplos, como ocorre o problema da exclusão mútua e apresente duas possíveis soluções.

EXERCÍCIO 03

13. O que são chamadas de sistema (SYSCALL)?
14. Qual a diferença entre uma chamada *trap* e uma interrupção?
15. Para um programados, uma chamada de sistema se parece com qualquer outra rotina de

EXERCÍCIO 03

biblioteca. É importante que um programador saiba quais rotinas de biblioteca resultam em chamadas de sistema? Sob quais circunstâncias e por quê?

16. A tabela da Figura 1 apresenta um comparativo entre a API em C Win32 e o API Unix POSIX. Considerando o que você sabe sobre chamadas de sistema, apresente um exemplo prático onde a comunicação entre processos é dificultada pela incompatibilidade da API Win32.

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

Figura 1: Comparação da API Win32 e Unix

17. Considere a chamada *read* apresentada em sala de aula. Descreva o comportamento do SO em cada passo de sua execução, assim como o fluxo dos dados na memória principal.

BIBLIOGRAFIA

SILBERSCHATZ, Abraham et al. **Operating system concepts**. Reading: Addison-Wesley, 1998.

TANENBAUM, Andrew S.; MACHADO FILHO, Nery. **Sistemas operacionais modernos**. Prentice-Hall, 1995.