

GS1018 – SISTEMAS OPERACIONAIS

Operating Systems – William Stallings – 7th Edition

Chapter 02 – Operating System Overview

Murielly Oliveira Nascimento – 11921BSI222 – murielly.nascimento@ufu.br

REVIEW QUESTIONS

2.1. What are three objectives of an OS design?

Um sistema operacional tem como objetivos a conveniência, torna o computador ou SO mais conveniente para uso; a eficiência, permite que os recursos do SO sejam usados de maneira mais eficiente; e a habilidade para evoluir e/ou ser atualizado.

2.2. What is the kernel of an OS?

O kernel, ou núcleo do SO, contém as funções usadas com mais frequência no sistema operacional.

2.3. What is multiprogramming?

Multiprogramação é, quando uma tarefa precisa esperar por uma operação de entrada/saída, o processador pode alternar para outra tarefa, que provavelmente não esteja esperando por uma operação de E/S.

2.4. What is a process?

Há diversas definições para processo, pode ser entendido, como um programa em execução; instância de um programa em execução em um computador; entidade designada para ser executada por um processador; unidade de atividade caracterizada por um único encadeamento sequencial de execução...

2.9. Explain the difference between a monolithic kernel and a microkernel.

Monolithic Kernel é implementado como um único processo, com todos os elementos compartilhando o mesmo espaço de endereço. Já a arquitetura microkernel delega apenas as funções essenciais ao kernel, comunicação entre processos (IPC), espaços de endereçamento e agendamento básico; e deixa as demais por conta de processos chamados servidores, que são executados no modo usuário e são tratados como qualquer outro aplicativo microkernel.

2.10. List the key design issues for an SMP operating system.

A arquitetura de hardware de computador SMP, apesar de suas vantagens, apresenta problemas, como agendamento, já que qualquer processador pode executar o agendamento, o que complica a tarefa de impor uma política de agendamento e garantir que a corrupção das estruturas de dados agendado seja evitada; sincronização, uma vez que com vários processos ativos tendo acesso a espaços de endereços compartilhados ou recursos de I/O compartilhados, deve-se tomar cuidado para fornecer uma sincronização eficaz; e processos

simultâneos concorrentes, as rotinas do kernel precisam ser reentrantes para permitir que vários processadores executem o mesmo código do kernel simultaneamente.

PROBLEMS

2.1. Suppose that we have a multiprogrammed computer in which each job has identical characteristics. In one computation period, T , for a job, half the time is spent in I/O and the other half in processor activity. Each job runs for a total of N periods. Assume that a simple round-robin scheduling is used, and that I/O operations can overlap with processor operation.

Define the following quantities:

- i) Turnaround time = actual time to complete a job;**
- ii) Throughput = average number of jobs completed per time period T ;**
- iii) Processor utilization = percentage of time that the processor is active (not waiting).**

Compute these quantities for one, two, and four simultaneous jobs, assuming that the period T is distributed in

each of the following ways:

a. I/O first half, processor second half

Primeiro caso:

- i) Turnaround time: TN
- ii) Throughput: 1 trabalho
- iii) Processor utilization: 50%

Segundo caso:

- i) Turnaround time: TN
- ii) Throughput: 2 trabalhos
- iii) Processor utilization: 100%

Terceiro caso:

- i) Turnaround time: $2*NT$
- ii) Throughput: 4 trabalhos
- iii) Processor utilization: 100%

b. I/O first and fourth quarters, processor second and third quarter

Primeiro caso:

- i) Turnaround time: T_N
- ii) Throughput: 1 trabalho
- iii) Processor utilization: 50%

Segundo caso:

- i) Turnaround time: T_N
- ii) Throughput: 2 trabalhos
- iii) Processor utilization: 100%

Terceiro caso:

- i) Turnaround time: $2 \cdot T_N$
- ii) Throughput: 4 trabalhos
- iii) Processor utilization: 100%

2.2. An I/O-bound program is one that, if run alone, would spend more time waiting for I/O than using the processor. A processor-bound program is the opposite. Suppose a short-term scheduling algorithm favors those programs that have used little processor time in the recent past. Explain why this algorithm favors I/O-bound programs and yet does not permanently deny processor time to processor-bound programs.

Embora um I/O-bound program tenha uma demanda pequena pelo processador ele ainda precisará dele eventualmente. Portanto, um I/O-bound program não será permanentemente negado acesso ao processador.

2.3. Contrast the scheduling policies you might use when trying to optimize a time-sharing system with those you would use to optimize a multiprogrammed batch system.

Em time sharing systems o processador é compartilhado entre os usuários, sendo ele utilizado para manipular 'jobs' iterativos. Nesse caso, é interessante aplicar políticas de divisão do tempo gasto em cada processo. Já em sistemas de multiprogramação o processador fica intercalando entre processos de I/O e outras tarefas. Logo, o ideal é trabalhar políticas que evitem tempo gasto nessas trocas.

2.4. What is the purpose of system calls, and how do system calls relate to the OS and to the concept of dual-mode (kernel-mode and user-mode) operation?

System calls é o meio pelo qual um processo solicita um serviço específico do Kernel. Existem várias delas que podem ser agrupadas em seis categorias: sistema de arquivos, processos, agendamento, comunicação entre processos, networking e miscellaneous.