GSI018 – SISTEMAS OPERACIONAIS

Operating Systems – William Stallings – 7th EditionChapter 01 – Computer System Overview

Pedro Henrique Silva Santana – 12011BSI218 – pedro.santana@ufu.br

Victor Hugo Martins Alves – 12011BSI217 – victor.alves1@ufu.br

Review Questions

2.1. What are three objectives of an OS design?

Os três objetivos do Sistema Operacional são ser conveniente para uso, eficiente e ter a habilidade para evoluir/ser atualizado (via “patchs” de sistema operacional).

2.2. What is the kernel of an OS?

O kernel é uma “porção” do sistema operacional alocado na memória principal que contém as funções mais frequentemente usadas no SO.

2.3. What is multiprogramming?

É o ato de expandir a memória para suportar 2 ou mais “job” e permitir o chaveamento da execução entre eles. Enquanto o “job” precisa esperar por I/O, o processador pode alternar para outro “job” em paralelo que provavelmente não estará a espera pelo I/O.

2.4. What is a process?

Processo possui diversas definições sendo alguma delas:

- definir um programa em execução;
- instanciar um programa em execução em um computador;
- entidade designada para ser executada por um processador;
- unidade de atividade com um único encadeamento sequencial de execução, um estado atual e recurso do sistema.

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

O **kernel monolítico** é definido por um kernel grande que inclui funções como agendamento, sistema de arquivos, rede, drivers de dispositivo, gerenciamento de memória, etc. normalmente este é implementado como um único processo, com todos os elementos compartilhando o mesmo espaço. Já o **microkernel** atribui apenas algumas funções essenciais, incluindo espaços de endereço, comunicação entre processos e agendamento básico. Os demais outros serviços são fornecidos por processos, que são executados no modo de usuário e tratados como qualquer outro aplicativo pelo microkernel.

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

Os principais problemas do SMP são:

- **Processos simultâneos**: rotinas do kernel devem ser reentrantes para garantir que vários processadores executem o mesmo código do kernel simultaneamente;
- **Agendamento**: qualquer processador pode executar o agendamento, dificultando a tarefa de impor uma política de agendamento e garantir que a corrupção das estruturas de dados seja evitada;
- **Sincronização**: com vários processadores ativos acessando espaços de endereços ou recursos I/O compartilhados, deve haver uma sincronização eficaz.

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).

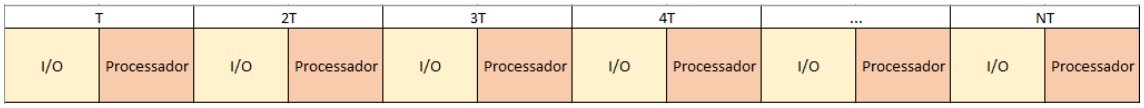
Compute these quantities for one, two, and four simultaneous jobs, assuming that the period T is distributed in each of the following ways:

a. I/O first half, processor second half

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

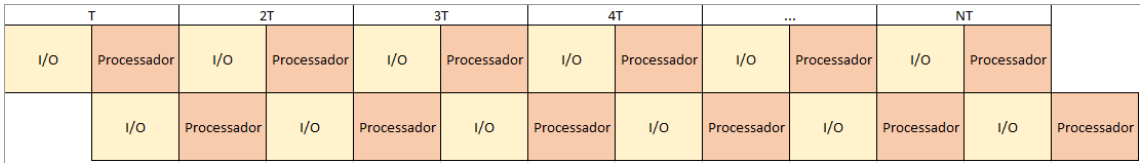
1 job

Turnaround = NT Uso do Processador: 50%



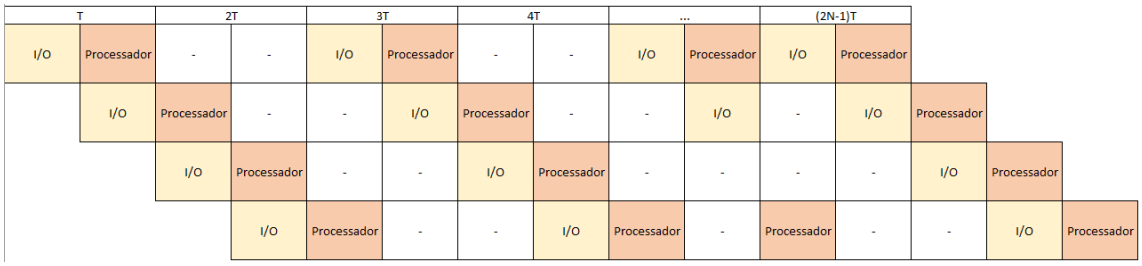
2 jobs

Turnaround = NT Uso do Processador: 100%



4 jobs

Turnaround = (2N-1)T Uso do Processador: 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.

O algoritmo irá favorecer processos relacionados ao I/O-bound pois estes utilizam menos tempo de processamento. Enquanto isso, o processo relacionado à CPU não será negado pois o mesmo irá para a fila e aguardará pelo processo em execução (I/O-bound).

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.

Comparando as políticas de agendamento entre o Time-Sharing System e o Multiprogrammed Batch System, para um grande número de “jobs”, é interessante que o usuário possa atuar diretamente com o computador, assim o Time-Sharing System tem uma vantagem pois permite que vários usuários acessem simultaneamente, por meio de terminais, e dividem o tempo de execução.

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?

O propósito das chamadas de sistema é efetuar uma requisição de serviço para o kernel para acesso da RAM. Quando um programa efetua uma chamada de sistema, o dispositivo muda do user-mode para kernel-mode, que por sua vez atende a requisição, retorna no processo e retoma o user-mode para que o usuário tenha o controle do dispositivo.