Noções superficiais de Sistemas Operacionais

Iuri de Carvalho Salgado

Inicialmente podemos estabelecer a evolução dos sistemas com fim operacional de uma máquina em cinco gerações:

* Geração 0: A primeira máquina de calcular com uma interface mecênica – Criada em 1942, La Pascaline foi projetada por Blaise Pascal.
* Geração 1: O Colossus – Idealizado por Alan Turin entre 1943 e 1945 foi um conjunto de computadores a válvula com grande importância no final da 2ª GM.
* Geração 2: TX-0 1956 (Transistorized Experimental computer zero) – Foi projetado no MIT e montado já com transistores.
* Geração 3: Época dos circuitos impressos integrados
* Geração 4: VLSI very large scale integration – Onde eram vendidas partes de um computador para serem montadas pelos usuários que, além disso precisavam escrever códigos para usa-los.

Os Sistemas Operacionais podem ser divididos para análise de algumas formas como a presença de Interface Gráfica ou apenas Linhas de Comando, se é um Software Livre ou Proprietário e quanto ao Gerenciamento de memória Física apenas como o CP/M e o MSDOS ou Gerenciamento de memória Física e Virtual como fazem os SOs atuais.

Estuturação do SO:

**Núcleo ou Kernel:**

Responsável por funções primordiais do sistema;

**Drivers:**

Comunicação com periféricos;

**Programas utilitários:**

Ferramentas do sistema;

**Códigos de carregamento do sistema.**

A virtualização é a capacidade de um SO gerenciar mais memória do que a máquina possui fisicamente!

Módulos de um SO:

**Gerenciador de Processos:**

Tarefas do computador

**Gerenciador de memória:**

Controle de memória

**Gerenciador de dispositivos:**

Condições para dispositivos serem usados por processos

**Gerenciador de arquivos:**

Organização, criação, leitura e permissões de arquivos

Para proteção do Sistema Operacional é separado um espaço de memória para o núcleo e um para os aplicativos de usuário. O design arquitetural é o seguinte:

|  |  |
| --- | --- |
| Núcleo de construção monolitica | Núcleo de contrução em camadas |
| * Todos os módulos estão no mesmo bloco * Cada parte do bloco pode ter finalidades e responsábilidades diferentes * Não existe a necessidade de protocolos de comunicação e politicas de segurança * Velocidade alta por ter acesso direto a regiões de memória diferentes * Programação mais complexa * Dependência entre diversas partes * Uma falha em uma das partes pode degenerar o núcleo rapidamente | * Camadas inferiores promovem serviços a camadas superiores * Quanto mais baixa a camada maior o envolvimento com hardware * Mudanças em uma macada são facilmente isoladas das superiores * Protocolo de comunicação para acesso estável, ordenado e com políticas de segurança para acesso a serviços providos pelas camadas inferiores * Dedicação de diferentes camadas a diferentes grupos de funções |

O **Gerenciamento de recursos** de hardware pode ser resumido a placa gráfica que solicita recursos de um monitor. Já os recursos de software gerenciados pelo SO tratam a solicitação de recursos da placa gráfica; A capacidade de uso da GUI com o motor gráfico com papel de manipulação de interfaces com criação de espaços de interação – janelas com objetos de interação dentro delas.

* Os recursos são limitados e o gerenciamento destes cabe ao SO, bem como o tempo de acesso a um recurso
* O conceito de processos e sub-processos é o modo commo um SO lida com os aplicativos, por meio de um identificador único
* O numero de registro de recurso são dados aos recursos do SO ou endereço de porto de entrada do recurso. O usuario não lida com isso, os aplicativos lidam atravez de chamas de sistemas

O **Gerenciamento de processo** gerencia o tempo em uso de cada processo da CPU. Um processo é um aplicativo sendo executado (job).

* O contexto de um processo refere-se aos recursos que estejam usando
* Os estados de um processo são **Novo –** processo em fase de criação, **Pronto –** preparado aguardando direcionamento para CPU, **Esperando –** aguardando recurso ou evento para execução, **Executando –** escalonado ou em execução na CPU e **Terminado –** contexto liberado.

Cada processo é representado por um bloco de controle de processos, o que consiste em um conjunto de metadados sobre os processos, normalmente referenciado por PCD (process control block) contendo informações sobre:

* Número do processo que o identifica.
* Estado atual do processo.
* Contador do programa que aponta o endereço de memoria que contém a proxima instrução do código de execução do aplicativo que deverá ser executado quando o processo mudar para o estado de executando
* Os registradores da CPU
* Dados de prioridade do processo
* O status de Entrada e Saída que inclui dados dos recursos usados no contexto

O **DEADLOCK** ocorre quando um processo A depende da liberação de um recurso que o processo B detem controle e este depende de um recurso sob controle do processo A, travando assim o sistema. A escolha de qual processo deve ir para o status de executando requer a estratégia de escalonamento ou política de escalonamento. Sendo elas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FCFS ou FIFO | SJF não preemptivo | SJF  preemptivo | Prioridade preemptivo | Round Robin |
| First come, first served ou first in, first out – Algoritmo não preemptivo. | Shortest job first – e executa o processo até seu termino | Shortest job first – e executa até terminar seu tempo | Conforme a prioridade dos processos obedecendo seu quantum | FIFO preemptivo – escalonamento por revezamento |

* Preemptivo refere-se a interrupção do processo sempre que o SO julgar necessário, como o tempo destidado ao processo (**quantum**).

O **Gerenciador de processo** pode definir arbitrariamente a prioridade de um processo. O quantum de um processo pode ser alterado pelo usuário, definindo novas prioridades e resultado na execução mais rápida de um processo. Usando uma linguagem de programação a comunicação entre aplicativos que ocorre atravez da API (application programing interface)

Referências:

**La Pascaline.** Wikipedia. Disponível em: https://pt.wikipedia.org/wiki/La\_pascaline. Acesso em: 18 de novembro de 2021.

**Colossus (computador).** Wikipedia. Disponível em: https://pt.wikipedia.org/wiki/Colossus\_(computador). Acesso em: 18 de novembro de 2021.

**TX-0.** Wikipedia. Disponível em: https://en.wikipedia.org/wiki/TX-0. Acesso em: 18 de novembro de 2021.

SOUZA,Osvaldo. **Sistema Operacionais.** CAPÍTULO 1 - AFINAL, PARA QUE SERVE OSISTEMA OPERACIONAL?. Ead Laureate.