

Luís Henrique Scantelbury de Almeida
Sistemas Operacionais I
15/12/2021

Lista de I/O

Faça um resumo sobre a vídeo-aula de Gerenciamento de I/O

Os sistemas operacionais são os responsáveis por interagir diretamente com o hardware, o qual é composto por uma grande parte de dispositivos de I/O (*Input & Output*). Para gerenciar esses dispositivos, o SO é dividido em diversas camadas. As mais baixas são responsáveis por abstrair os diferentes componentes de hardware para facilitar a interação dos mesmos com as camadas mais altas. Essas camadas são classificadas como: Device Drivers (mais conhecidos como drivers) e Subsistema de I/O.

- **Device Driver**: Se comunica com o dispositivo a nível de hardware através de **controladores** específicos para o aparelho. Um driver se comunica com apenas um tipo de dispositivo, e deve ser escrito para cada tipo de sistema operacional. Estes são geralmente escritos em **assembly** e são acoplados ao núcleo do SO.
- **Subsistema I/O**: Gerenciam as funções traduzidas pelas camadas mais baixa, tratando-as de forma mais generalizada independente do hardware acoplado. Também é responsável pelo tratamento de erros, segurança do acesso aos dispositivos e pela **bufferização** de comandos de **I/O**, que consiste na alocação de um certo número de operações de **I/O** numa memória intermediária.

Descreva as principais funções do subsistema de I/O

Processar as informações passadas pelas camadas inferiores, independente do hardware acoplado e passá-las para as camadas superiores do SO. Tratar erros. Relacionar o dispositivo em questão com seu respectivo driver. Proteger o acesso aos dispositivos em uso. Armazenar operações de **I/O** numa memória intermediária para reduzir o número de operações no futuro.

Um Device Driver desenvolvido para um SO pode ser usado em outro? Por que?

Não, pois cada SO interage de forma diferente com seus drivers, por isso drivers de um mesmo dispositivo devem ser desenvolvidos especificamente para o sistema operacional em foco.

Pesquise sobre o desenvolvimento de device drivers para Linux e descreva as principais atividades do desenvolvedor.

Por ser um código livre, o desenvolvimento de drivers para Linux, requer apenas conhecimento e habilidade por parte do desenvolvedor, o qual programa o código do driver especificamente para o kernel onde o driver será acoplado, o integra, compila e gera uma nova build do sistema operacional. O Linux tem SDKs específicos para operações onde o SO é mais utilizado, como os device drivers de rede, que são específicos para as tarefas de computação distribuída em rede, prática recorrente no meio Linux.

Pesquise sobre o desenvolvimento de device drivers para Windows e descreva as principais atividades para o desenvolvedor.

Por não ser de código livre, o desenvolvimento de drivers para Windows requer o kit de criação de drivers (**Windows Driver Kit**) disponibilizado pela microsoft, o subsistema para drivers nativo (**Windows SDK**), o editor de código para projetos Windows (**Microsoft Visual Studio**) e um compilador para o código que vem contido no WDK, sendo este para código C/C++