

QUESTÃO 2 - SISTEMAS DE E/S.

Ao comparar o modelo de *device drivers* implementado no **Windows** com o **Linux**, a primeira e principal diferença é que o cerne do **Linux** é distribuído com os próprios drivers, enquanto que o do Windows não inclui drivers, muito pelo contrário: Os drivers de dispositivos do Windows são “gravados” utilizando o *Windows Driver Model* (WDM), que oferece suporte para que os drivers possam ser carregados e descarregados de acordo com a necessidade. O WDM fornece três camadas para os drivers, na qual informações entre o hardware e o dispositivo de E/S são trocadas. [1]

Em relação a segurança, todos os drivers de dispositivo do Windows precisam de uma assinatura digital para serem utilizados. Estas podem ser assinadas, mas as ideais são as com certificado válido pela Microsoft. O kernel do Linux também pode ser configurado para verificar as assinaturas dos módulos do kernel que estão sendo carregados e desaprovar os não confiáveis, mas a rigidez destas validações é customizável e flexível. [2][3]

QUESTÃO 3 - ARQUIVOS DE SISTEMA.

Inode

Em sistemas operacionais baseados em Unix, incluindo o próprio Linux, é utilizado um método de alocação no qual cada “objeto” (arquivo ou diretório) do *File-system* é descrito por um *Indexed Node*, ou **Inode**, como é chamado comumente [4]. Cada Inode é identificado por um número inteiro e único, e armazena diversas informações sobre o objeto sendo representado, entre elas: [5]

- **Formato de arquivo.** Se é um diretório, arquivo normal, etc.
- **Horário.** Última vez que foi acessado ou modificado.
- **Permissões** de leitura, escrita e/ou execução, para diferentes usuários e grupos.
- **Tamanho do arquivo.**
- **Localização do arquivo** no disco.
- **User ID**, que referencia o dono do arquivo.

Os Inodes são armazenados em uma região específica e conhecida do disco, no qual o tamanho é definido na inicialização do sistema, limitando a quantidade máxima de Inodes possíveis. Para isto, existe uma prática comum de reservar 1% do total do disco para tal uso. Há uma tabela de Inodes onde eles são indexados de acordo com seus números-identificadores, que permite o driver do sistema de arquivos do kernel acessar o conteúdo do inode, incluindo a localização do arquivo, permitindo assim o acesso ao arquivo. [6]

FAT

Como alternativa mais simples, o **FAT** é popular entre dispositivos de armazenamento de propósito geral, e é suportado por grande parte dos sistemas operacionais, proporcionando performance e robustez adequada para maior parte das situações.

A tecnologia consiste majoritariamente de uma tabela de índices, alocada na inicialização do sistema, que possui vários “registros” que referenciam *clusters* - áreas contíguas de armazenamento no disco. Cada um destes registros aponta para o próximo, ou indica o fim de uma região de armazenamento, permitindo que o sistema operacional leia o dispositivo de armazenamento similarmente a como se atravessa uma estrutura de lista-encadeada, verificando cada cluster até achar seu alvo [7].

Levando em consideração todos os Clusters no sistema FAT, podemos dividi-los em 3 grandes conjuntos, denominados *setores* ou *regions*: *Boot Sector*, *FAT Region* e *Data Region*. Estes três estão presentes em qualquer versão do sistema FAT, mas em versões de 12 e 16 bits, há também o *Root Directory Section* [8]. Detalhando mais sobre cada região:

- **Boot Sector.** Normalmente localizado no começo do sistema de arquivos, contém todo os arquivos e códigos necessários para inicializar o sistema operacional. Por isto, é considerado um setor *reservado*.

- **FAT Region.** Contém a tabela que mapeia os *clusters* presentes na *Data Region*. Pode possuir cópias extras, mas que são raramente utilizadas fora de uma situação de backup
- **Data Region.** Ocupa maior parte da partição, e é onde todos os dados de arquivos e diretórios são guardados. Em versões abaixo do FAT12, arquivos do *Diretório Root* são guardados aqui, mas em versões acima, há uma região específica para eles - a *Root Directory Region*.

REFERÊNCIAS

1. Windows Documentation, [Microsoft. Windows Driver Model \(WDM\)](#). Acesso em 10/06/2020.
2. Websetnet, Stuart, Aaron. [Linux vs. Windows device driver model: architecture, APIs and build environment comparison](#). Acesso em 10/06/2020.
3. Kernel Documentation, Linux Foundation. [Driver Model](#). Acesso em 11/06/2020.
4. BACH, Maurice J. [The design of the Unix operating system](#). Upper Saddle River: Prentice Hall, 1990.
5. LinOxide. Zachariah, Bobbin. [Detailed Understanding of Linux Inodes with Example](#). Acesso em 15/06/2020.
6. TANENBAUM, [Andrew S. Modern Operating Systems](#). Pearson. Capítulo 4.
7. https://en.wikipedia.org/wiki/File_Allocation_Table. Acesso em 15/06/2020.
8. <https://www.win.tue.nl/~aeb/linux/fs/fat/fat-1.html>. Acesso em 15/06/2020.