Sistemas Operacionais

Gladimir - [gladimir@gmail.com](mailto:gladimir@gmail.com)

07/03/2023

Coisas que vamos aprender:

* Como funciona um sistema operacional
* Princípios básicos de eletricidade
* Identificar a estrutura interna
* Cálculos de diferentes sistemas de numeração
* Organização de memória
* Organização de arquivos
* Conceitos de threads
* Livros:
  + Sistemas operacionais modernos
  + Organização e estrutura de dados

Endereço relativo

* está em relação a pasta atual

Endereço absoluto

* caminho inteiro

Aula 2 - 13/03/23

Filme depois exercícios

1) Linux é um sistema operacional que surgiu como resposta aos softwares pagos (windows).

2) Ele explica que as pessoas usam programas em seus computadores (não Sistemas Operacionais) e o único papel do SO é ajudar esses programas a funcionarem.

3) Stallman pensou no kernel como um sistema modularizado que comunica consigo mesmo por meio de um protocolo, já Linus criou um monolito, um bloco de software (monolito) que evitava diversos bugs gerados pela comunicação entre diversas partes.

4) Pois tiveram dificuldades em desenvolver o kernel, coisa que foi feita primeiro e melhor por Linus

5) Software livre pode ser ou não gratuito, porém o termo, segundo Stallman, tem relação com a capacidade de distribuição e não com o preço.

6) No filme é citado especialmente o suporte como forma de monetizar o software livre.

7) O GNU usa o linux como kernel e os dois juntos formam o GNU/linux, o Sistema Operacional.

8) O fator crucial foi ele ser livre e aberto para mudanças conforme sua necessidade.

9) As pessoas confundiam "livre" como grátis, o que não era o caso, pois o "livre" tem relação com a liberdade de distribuição.

10) Software livre é software com liberdade de distribuição, freeware é software grátis, enquanto código aberto significa que seu código fonte é passível de mudanças conforme a necessidade.

11) Esta discussão era relacionada à divisão do software em núcleos menores, ou o uso do chamado monolito. A primeira tem como pontos fortes a modularização do software, porém ao custo de estar aberta a bugs relacionados a comunicação, já o segundo ao passo que evita estes bugs talvez não seja um aproveitamento tão moderno em questão de projeto de software. Dito isto, acredito que a modularização se bem feita pode ajudar a organização do código e evitar que partes não usadas sejam processadas.