
MAC0422 - Sistemas Operacionais

Daniel Macêdo Batista

IME - USP, 10 de Setembro de 2020

Roteiro

Um pouco de
história

Hardware

Um pouco de história

Hardware

▷ Um pouco de
história

Hardware

Um pouco de história

Como foi a evolução?

Um pouco de
história

Hardware

- De um modo geral, muito casada com o hardware (se o hardware não muda, para que mudar o SO?)
- Desconsideraremos as tentativas de uma máquina de calcular 100% mecânica

Válvulas (1945-1955)

Um pouco de
história

Hardware

- Uma válvula é um dispositivo, de forma grosseira, que permite o fluxo de corrente elétrica em um único sentido, quando a tensão em determinados terminais da válvula atendem uma certa propriedade.
- Útil para amplificadores e podem servir também como chaves “inteligentes”
- Com essas chaves “inteligentes” é possível realizar operações lógicas que são a base para um computador!
 - Z3 na Alemanha
 - Colossus na Inglaterra
 - Mark I e ENIAC nos EUA

Válvulas (1945-1955)

Um pouco de
história

Hardware

- Nessa época: máquinas gigantes, programação apenas com linguagem de máquina.
- Não havia SO. A gerência do recurso (a máquina) era feita de forma convencional: Se você quer usar a máquina, encontre um slot livre nessa agenda, entre na sala onde ela está, ponha as chaves para cima e para baixo e use no seu tempo (Torcer para alguma válvula não queimar)
- Melhorou um pouco nos anos 50 com os cartões perfurados. Mais fácil para passar seu programa para a máquina.

Transistores e sistemas em lote (1955-1965)

Um pouco de
história

Hardware

- Em essência o transistor faz o mesmo que a válvula mas ocupando muito menos espaço, dissipando bem menos calor e consumindo menos energia (mais difícil de dar defeito)
- Computadores deixam de ser só projetos de pesquisa em universidades ou nas forças militares. Agora empresas fabricavam e vendiam os chamados *mainframes*, embora ainda fossem máquinas muito caras.
- Começa a haver uma separação entre projetista, construtor, operador, programador e mantenedor do computador. Uma pessoa não precisa entender tudo como acontecia antes.
- Máquinas ainda grandes, separadas em salas especiais bem frias

Transistores e sistemas em lote (1955-1965)

Um pouco de
história

Hardware

- O mainframe rodava *jobs*. Um *job* era um programa que era escrito por um programador, provavelmente na linguagem FORTRAN, e que era enviado para cartões perfurados. O programador tinha que levar seus cartões para uma sala de entrada e passar para o operador que ia colocar os cartões no mainframe. Isso podia demorar!
- O operador tinha que carregar o compilador de FORTRAN no mainframe e depois os cartões do programador. Se fosse outra linguagem, carregar outro compilador.
- Depois que o programa terminava de executar, o operador colocava na sala de saída as páginas da impressora com a saída do programa.

Transistores e sistemas em lote (1955-1965)

Um pouco de
história

Hardware

- ❑ Como otimizar? Com o sistema de lote. Carrega todos os jobs (cartões perfurados) de FORTRAN por exemplo, lê e salva em uma fita magnética em um equipamento mais barato que só fazia isso, e depois enviava a fita magnética para o mainframe, já carregado com o compilador FORTRAN.
- ❑ A saída era guardada em fita magnética também e só depois tudo era impresso (salvar em fita era mais rápido que imprimir)
- ❑ Começa a haver precursores do SO. O mainframe agora lê vários programas que estão na fita, um depois do outro mas muitas das tarefas do SO ainda estão sendo feitas manualmente (múltiplos processos brigando)
- ❑ Não existia cientista da computação. Físicos e engenheiros usavam os computadores.

Circuitos integrados e multiprogramação (1965-1980)

Um pouco de
história

Hardware

- Agora, transistores menores e encapsulados em CIs (circuitos integrados)
- Dois tipos de máquinas: os computadores científicos para cálculos pesados e os computadores orientados a caracter para armazenamento e impressão de dados principalmente.
- **IMPORTANTE** Aparece a habilidade da memória possuir vários programas na memória uma única vez e se um programa realiza operação de E/S, o processador pode ficar livre para ser usado por outro programa
- **IMPORTANTE** Aparece também o Spooling (SPOOL – Simultaneous Peripheral Operation On Line) com a habilidade de carregar cartões perfurados direto no disco

Circuitos integrados e multiprogramação (1965-1980)

Um pouco de
história

Hardware

- **IMPORTANTE** Aparece o compartilhamento de tempo: cada usuário com um terminal. Se há tempo desperdiçado por um usuário, pode ser usado por outro.
- Minicomputadores aparecem (US\$ 120.000 por um. Sim, isso era barato! Antes um computador custava mais de US\$ 2.000.000)
- Os primeiros SOs aparecem: MULTICS → UNIX
- Com o código aberto, muitas empresas desenvolvem suas versões do UNIX. Destaque para o System V (AT&T) e para o BSD (Berkeley). A IEEE define um padrão para permitir que códigos rodem em qualquer UNIX: POSIX (uma interface mínima de chamadas de sistema). Mais tarde, em 1991, o Linux vai surgir!

Computadores pessoais (1980-presente)

Um pouco de
história

Hardware

- com a integração de larga escala em circuitos integrados, milhares de transistores ocupam 1 cm^2 nos CIs. Mais transistores → maior capacidade de realizar operações lógicas, mais memória e a um preço menor
- Intel lança o processador 8080. Um SO torna-se necessário
 - CP/M (Control Program for Microcomputers) surge
 - IBM lança o IBM PC e quer programas para rodar nele (interpretador BASIC de Bill Gates em particular)
 - IBM PC também precisa de SO (após um monte de “passadas de perna” o IBM PC é lançado com o MS-DOS e com o interpretador BASIC)
- 80286 → 80386 (GUI e Windows – Assistam Piratas do Vale do Silício) → 80486 → Pentium → ...

Um pouco de
história

▷ Hardware

Hardware

Alguns utilitários e arquivos no GNU/Linux

Um pouco de
história

Hardware

- `lshw`
- `free`
- `strace <comando>`
- `cat /proc/cpuinfo`
- `ls /dev`

Relembrando sobre a arquitetura de Von Neumann

Um pouco de
história

Hardware

- CPU (1 ou mais)
- Memória
- Dispositivos de E/S (pelo menos um teclado, um monitor e um disco rígido)
- Mas como fazer a comunicação entre eles? Barramento (aquele bus lá do /dev tem relação com isso!)

- Loop:
 1. obter instruções da memória
 2. decodificar a instrução (tipo e operandos)
 3. executar a instrução
 4. repetir para a próxima instrução
- Isso para diversos programas, muitos podem estar concorrendo por esse “serviço”
- Registradores aceleram as ações (mudanças de contexto)
- Conjuntos de instruções são específicos de cada processador

- Como acelerar a execução?
 - Pipeline (já existia manualmente há décadas atrás)
 - Unidades em sequência para várias tarefas. Enquanto executa uma instrução, já está indo na memória carregar a próxima, o mesmo enquanto decodifica
- Mas já que as instruções podem ser diferentes...
 - CPU superescalar
 - Unidades diferentes para cada tipo de instrução. Enquanto executa uma de aritmética inteira, pode executar também uma de aritmética de ponto flutuante (Claro que aqui faz sentido se tem mais de um pipeline)

- Se consegue miniaturizar tudo
 - Mais de uma CPU ou núcleo
 - Várias unidades de processamento para rodar mais de 1 thread por vez ou mais de 1 processo por vez

Como fazer a ligação entre o programa, o SO e a CPU?

Um pouco de história

Hardware

- ❑ Registrador PSW (Program Status Word) – Bits de controle como modo kernel e modo usuário
- ❑ Quando em modo usuário, apenas um subconjunto das instruções do processador são permitidos para um programa
- ❑ Se precisa de um serviço do SO, é necessário fazer uma chamada de sistema

```
man syscalls
/usr/include/x86_64-linux-gnu/bits/syscall.h
```