



Arquitetura de Computadores e Redes

Dia 1 - Arquitetura de Computadores e SO

Índice

1. Alan Mathison Turing
2. Arquitetura de Computadores
3. CPU
4. Memórias
5. Sistema Operacional
6. Sistema de Arquivos
7. API SO

Alan Mathison Turing



Breve resumo

- 1912 (Londres) - 1954 (Cheshire)
- Na década de 1930, criou a classe de máquinas que concebeu o estudo da **computabilidade** e estabeleceu os **limites** da ciência da computação.
- Nenhum outro tipo de máquina proposta **posteriormente** tem maior poder computacional. Mesmo se considerarmos os computadores atuais!
- Em 1936, formalizou o conceito de **Algoritmo** na tese de Church-Turing.

Alan Mathison Turing



Alan Mathison Turing

Filme

- The Imitation Game (2014) - <https://www.imdb.com/title/tt2084970/>

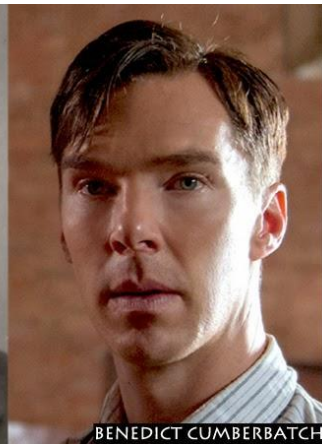


IMDb Menu All Search IMDb

FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE | SHARE

+ **O Jogo da Imitação (2014)** ★ 8,0¹⁰ 675.054 ☆ Rate This

The Imitation Game (*original title*)
12 | 1h 54min | Biography, Drama, Thriller | 5 February 2015 (Brazil)



Alan Mathison Turing

Prêmio Alan Turing

- Concedido anualmente pela Associação para Maquinaria da Computação.
- As contribuições da pessoa selecionada devem ser duradouras e fundamentais.
- Considerado o prêmio Nobel da computação.
- Alguns ganhadores: Criador da linguagem C, Linux, Internet...

Alan Mathison Turing

Prêmio Alan Turing

- <https://amturing.acm.org/>



Alan Mathison Turing

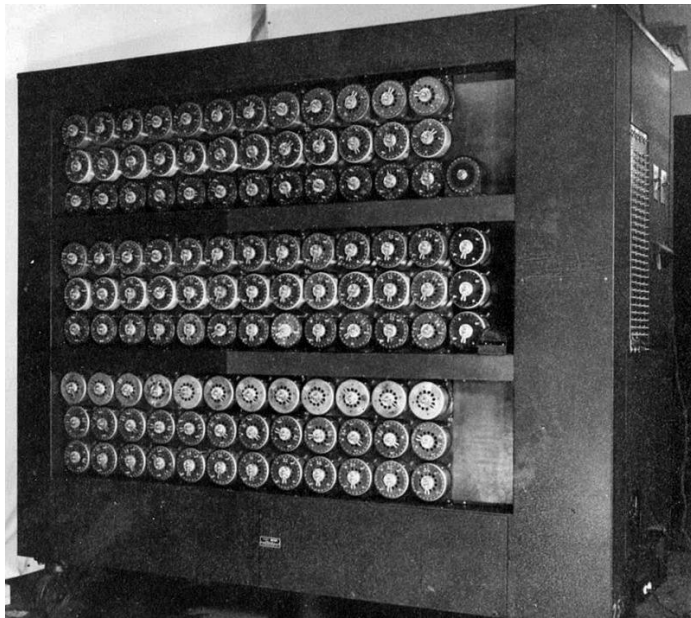
Teste de Turing



Alan Mathison Turing

Teste de Turing

- [Publicação - 1950](#)



MIND

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY



I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can machines think?' is to be sought in a statistical survey such as a Gallup

Alan Mathison Turing

Teste de Turing



09/06/2014 21h20 - Atualizado em 09/06/2014 21h20

Computador convence juízes de que é garoto de 13 anos em 'teste de Turing'

De 30 juízes, 10 acharam que 'Eugene Goostman' era humano.
Software é um 'chatbot' que começou a ser desenvolvido em 2001.

Alan Mathison Turing

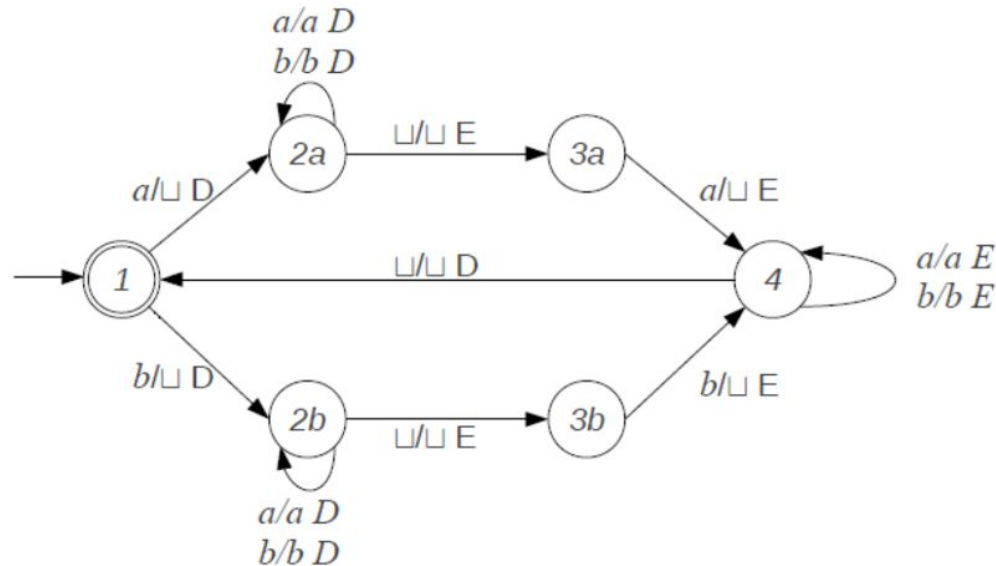
Contribuições fundamentais:

- Máquinas de Turing
 - Dispositivo teórico, um modelo abstrato de computador, com uma capacidade de memória infinita.
 - Dispositivo imaginário que formou a estrutura para fundamentar a computação moderna.
- Definição de Algoritmo
 - Um conjunto finito de instruções simples e precisas, que são descritas com um número finito de símbolos.
 - Se é possível escrever um algoritmo, em qualquer linguagem, é possível também desenhar uma máquina de Turing equivalente.

Alan Mathison Turing

Máquinas de Turing

MT que reconhece palíndromos de tamanho par sobre o alfabeto $\{a,b\}$. Exemplos: *abba*, *baab*, *aabbaa*, *bbaabb*...



Alan Mathison Turing

Máquinas de Turing



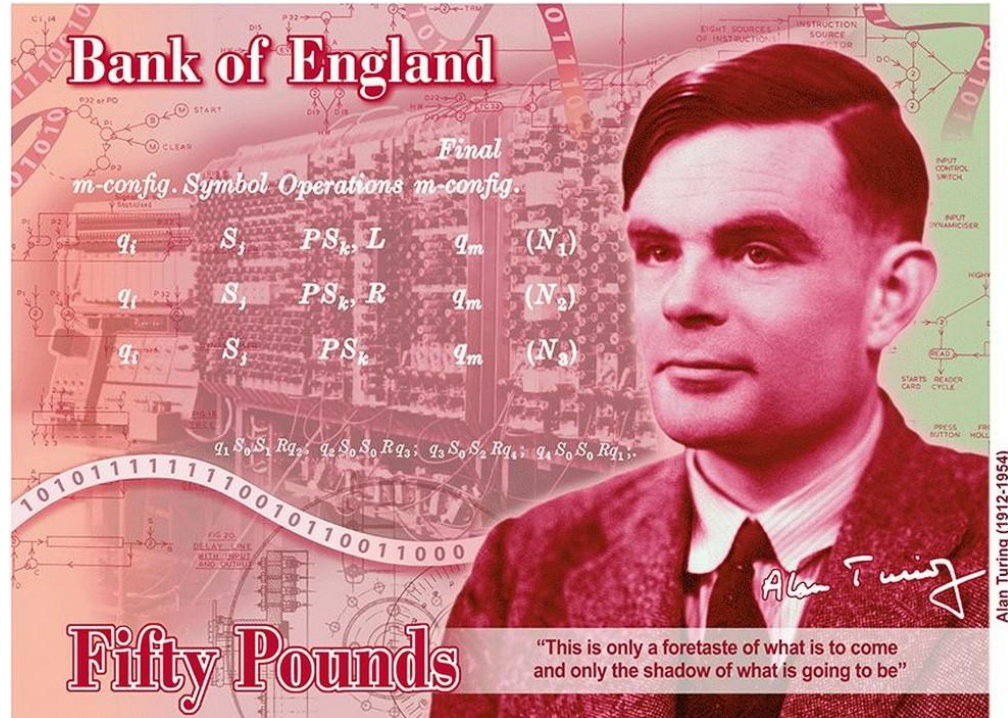
Alan Mathison Turing

Ele foi um dos responsáveis por descobrir o local exato onde as tropas nazistas estariam em 6 de junho de 1944, que culminou no desembarque de 155 mil soldados aliados na Normandia.



Alan Mathison Turing

Nota de 50 libras



Alan Mathison Turing

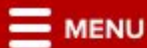
Lei de Turing

Somente no ano de 1967, as leis do Reino Unido deixaram de considerar a homossexualidade crime.

Entretanto, somente em 2017 uma lei cancelou a condenação de todos aqueles homens que foram injustamente perseguidos antes do fim da legislação homofóbica.

Cerca de 15 mil dos 65 mil condenados naquele tempo ainda estão vivos. A lei foi batizada com o nome da mais famosa vítima da homofobia institucionalizada na Inglaterra.

Alan Mathison Turing



TECNOLOGIA E GAMES

23/12/2013 22h55 - Atualizado em 24/12/2013 07h21

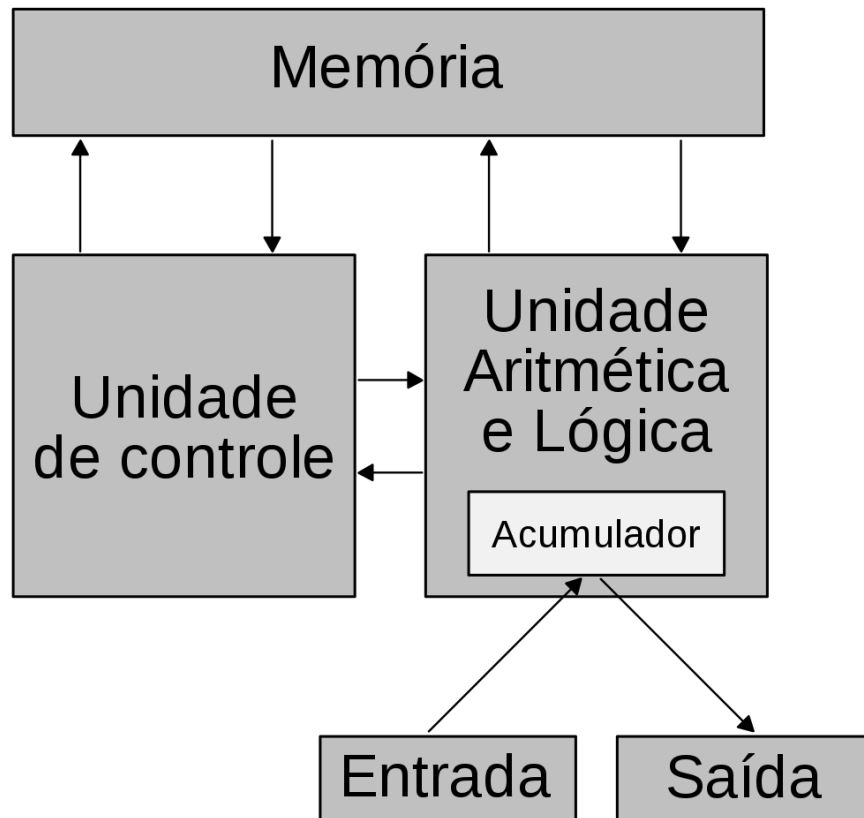
Pai da computação, Turing recebe o perdão real 59 anos após morrer

Alan Turing (1912-1954) foi condenado pelo Reino Unido por ser gay. O matemático decodificou as mensagens alemãs na 2ª Guerra Mundial.

A morte de Turing foi classificada a época como suicídio, mas estudiosos e biógrafos concluíram que o matemático faleceu de intoxicação devido aos remédios que tomava para cumprir sua pena.

Arquitetura de Computadores

- Arquitetura de Von Neumann



Arquitetura de Computadores



Arquitetura de Computadores

- Von Neumann aplicou os princípios de Turing e, em junho de 1948, o mundo teve a primeira demonstração prática de um computador.
- Ainda em 1948, Von Neumann, membro do Instituto de Estudos Avançados em Princeton, convidou Alan Turing para ser diretor do laboratório de computação da Universidade de Manchester, e ele aceitou o convite.
- Turing deu prosseguimento aos seus estudos mudando o foco para Inteligência Artificial, tendo sido o pioneiro no assunto ligando a abstração de sistemas cognitivos à realidade dos números.

Arquitetura de Computadores

- Código de alto nível vs Código de baixo nível
- Processo de interpretação / compilação traduz o código em alto nível de abstração para N outras linguagens com menor grau de abstração (baixo nível)
- Quanto menor o nível de abstração mais próximo a linguagem estará de uma sequência de operações aritméticas simples, como soma, multiplicação, etc.

Arquitetura de Computadores

Um programa escrito em linguagem de baixo nível é composto de um conjunto de instruções **simples** que podem ser executadas pelo processador quase que diretamente da forma como estão escritas.

- Quanto **maior** o nível de abstração da linguagem, mais **facilitado** é o aprendizado, com menor custo de elaboração e manutenção de software.
- Quanto **menor** o nível de abstração da linguagem, mais **dificultado** é o aprendizado, porém com um melhor aproveitamento do hardware e maior velocidade de processamento.

Arquitetura de Computadores

- Exemplos de tradução, alto nível para baixo nível, em linguagens compiladas:

Arquivos:

○ .java	->	.class
.cpp	->	.c
.cs -> .fs		

Arquitetura de Computadores

- **Arquitetura de Von Neumann**

1. Um programa é escrito já em linguagem baixo nível ou então é "convertido", compilado ou interpretado, para uma linguagem de baixo nível.

A partir daí a linguagem é convertida para linguagem de máquina, que é a linguagem que de fato o computador "fala". Essas são instruções bem simples como "some A com B", "mova A para o registrador X" e elas são convertidas para números binários.

Arquitetura de Computadores

- **Arquitetura de Von Neumann**

2. A partir daí essas instruções são carregadas na memória principal, a memória RAM dos computadores. A memória principal armazena as informações de maneira temporária, enquanto tiver energia passando por ela, ela consegue representar essas informações, porém, quando o computador é desligado, as informações são perdidas.
2. As informações, então, são buscadas na memória principal pela CPU, que trabalha em um loop buscando informações nela, processando e devolvendo as saídas dos processamentos para a memória.

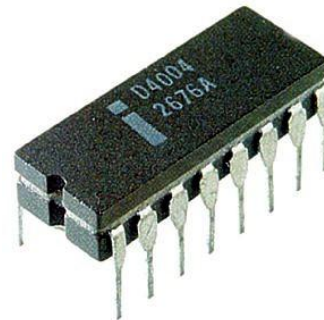
Arquitetura de Computadores

- Simulação de hardware

buthowdoitknow.com/but_how_do_it_know_cpu_model.html

Arquitetura de Computadores

- CPU



O primeiro processador foi o *Intel 4004*, que foi desenvolvido em uma arquitetura 4 bits, isso quer dizer, de maneira simples, que ele era capaz de realizar operações com dados de 4 bits cada.

Esse processador possuía os barramentos de dados e de endereços, que são aquelas vias que transmitem os dados que vimos no conteúdo, capazes de trafegar 4 bits, ou seja, com 4 fios.

Arquitetura de Computadores

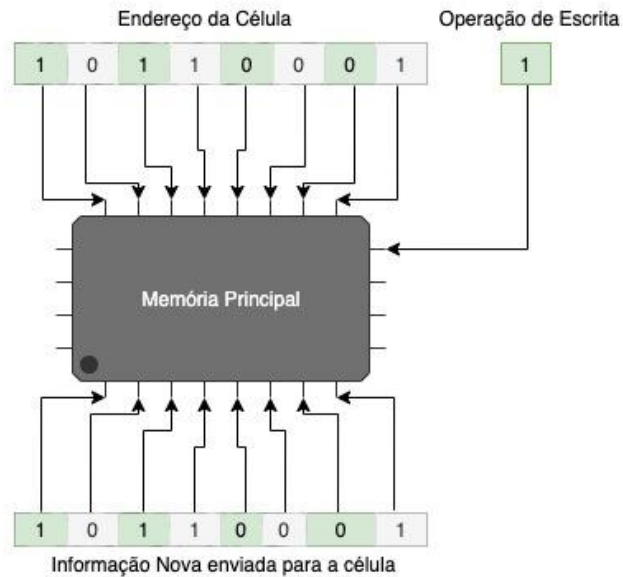
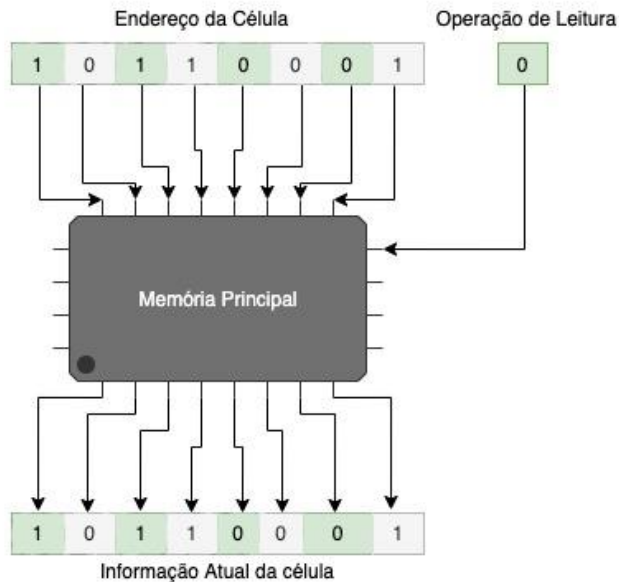
- CPU



Depois disso tivemos CPUs 8-bits, elas foram utilizadas nos primeiros PCs que rodavam o DOS e nos famosos *game-boys*, aqueles videogames portáteis dos anos 80 e 90.

Nessa arquitetura, as CPUs podiam operar dados de 8 dígitos binários, parecida com esse exemplo que vimos no conteúdo.

Arquitetura de Computadores



Arquitetura de Computadores

- CPU

Rapidamente as CPUs evoluíram para CPUs 16-bit e, então, 32-bit. Essas últimas arquiteturas, foram bem populares e estiveram presentes em várias máquinas.

Com uma arquitetura 32-bit, é possível trafegar pelos barramentos 32 bits de dados e a partir disso temos uma curiosidade bem legal: se tivermos somente um dígito, poderíamos endereçar duas células, uma célula 0 e outra 1.

Arquitetura de Computadores

- CPU

Com dois dígitos conseguimos endereçar 4 células: 00, 01, 10 e 11 e assim por diante. Podemos chegar nesse valor elevando 2, duas opções 0 ou 1, ao número de bits possíveis, por exemplo 2 elevado a 1, ou 2 elevado a 2, nos exemplos anteriores.

Com a arquitetura de 32 bits, temos 32 dígitos, dessa forma conseguiríamos ter 2 elevado aos 32 dígitos. **Vamos ver a quantos endereços possíveis esse número equivale.**

Arquitetura de Computadores

- CPU

Chegamos em **4.294.967.296** endereços possíveis, parece muito né?!

Mas agora vamos imaginar que cada endereço possui uma célula capaz de armazenar um byte, ou seja, 8 bits. Dessa forma, poderíamos ter no máximo 4.294.967.296 bytes de memória sendo utilizados. Vamos melhorar a leitura desse número, como a cada 1024 byte temos um kilobyte, vamos dividir por 1024, em seguida por 1024 novamente para ter o valor em megabytes e novamente para ter o valor em gigabytes.

Arquitetura de Computadores

- CPU

Chegamos em **4096 MB**, ou **4GB**, isso quer dizer que essa arquitetura, de 32 bits, possui esse limite de bytes para a memória RAM. E o que isso significa? Que, por mais que seja adicionado mais do que isso de memória RAM, não é possível endereçar as células a mais, não sendo possível utilizá-la. **Ou seja: máquinas 32 bits não utilizam mais que 4GB de memória RAM. Elas não conseguem endereçar mais que isso!**

Hoje muitas máquinas possuem mais do que 4GB de memória e, por isso, a arquitetura de 64 bits é bem popular nos computadores. Vamos ver a capacidade de endereçamento dessa arquitetura.

Arquitetura de Computadores

- CPU

Para processadores de 64 bits:

2^{64} .

Chegamos em um número perto dos **17 bilhões de gigabytes**, um número que ainda estamos *um pouco* longe de precisar utilizar em memória RAM.

Arquitetura de Computadores

- **Memórias**

Temos as memórias **primárias**, que são memórias mais rápidas, porém, que não persistem os dados quando desligadas.

E temos também as memórias **secundárias**, que são capazes de fazer essa persistências, porém são muito mais lentas para leitura e escrita.

Arquitetura de Computadores

- Memórias



Arquitetura de Computadores

- **Memórias**

O importante é vermos como o processo de leitura e escrita é realizado totalmente de maneira **mecânica**: esse braço que vocês estão vendo se mexer funciona através de eletromagnetismo e é responsável por gravar os bits (1 e 0) no disco, que está rodando bem rápido. **Ele magnetiza trechos do disco**. Um tipo de magnetização é lida como 1 e outro tipo é lida como 0.

O disco possui diversas trilhas e setores onde os bits são gravados, tanto a leitura como a escrita é feita através da cabeça magnética, que fica na pontinha do braço.

Arquitetura de Computadores

- **Memórias**

Por mais que pareça que a cabeça está encostando no disco ela não está! Porém, ela realmente está muito perto, o que é um dos motivos para esse componente ser tão sensível. Se essa cabeça encostar no HD irá danificá-lo.

E se ela não está encostando, como que os dados são lidos e escritos?!

Arquitetura de Computadores

- **Memórias**

Esse processo é feito pela cabeça, através de sinais elétricos contra o disco. Conforme o disco vai girando e a cabeça se movimentando, a cabeça chega até a posição desejada e então ela emite um sinal que é gravado em uma película magnética que fica sobre o disco.

Quando vocês acharem que um arquivo está demorando para ser copiado lembre-se disso!

Arquitetura de Computadores

- **Memórias**

Quando deletamos algo ou esvaziamos a lixeira, estamos apagando apenas os endereços para esses arquivos no Sistema Operacional. Afinal o disco não foi de fato formatado (formatação raiz).

Os arquivos ainda se mantêm lá, sendo **perfeitamente possível recuperá-los**.

Arquitetura de Computadores



Arquitetura de Computadores

- Memórias

SSD



Arquitetura de Computadores

- **Memórias**

O **SSD** ainda é **lento em relação a RAM**, mas é muito mais rápido do que o HD.

Isso se dá porque eles possuem memória **flash**, bem parecido com a que temos em pendrives e em cartões de memória, e não precisa de toda aquela mecânica de disco e braço para a leitura e gravação. Dessa forma, é muito rápido armazenar e ler informações nesses dispositivos.

Porém, como vocês devem imaginar, esses dispositivos **são bem mais caros** que um HD convencional. Lembrem-se: quanto mais rápido, mais caro!

Arquitetura de Computadores

- **Memórias**

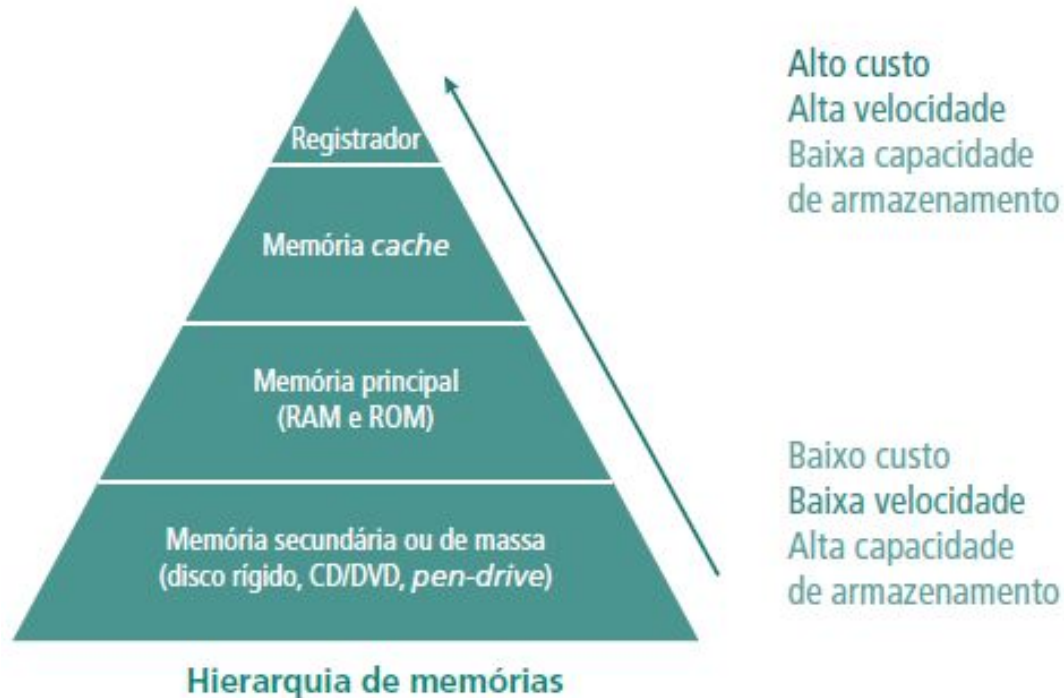
Um mundo ideal seria ter memórias grandes, rápidas, não voláteis, ou seja, que não perdem os dados ao desligá-las e tudo isso com baixo custo.

Mas sabemos que essa não é a realidade, na verdade trabalhamos com diversos processos que precisam compartilhar entre si os recursos disponíveis.

Porém, conseguimos lidar muito bem com esse cenário utilizando **tipos de memória diferentes para usos específicos**, conseguindo balancear essa situação.

Arquitetura de Computadores

- Memórias



Arquitetura de Computadores

- **Memórias**

Percebam que no topo da pirâmide temos os **registradores**, que são as memórias de maior velocidade, pois além de serem componentes bem rápidos, elas se localizam dentro da CPU, porém, a capacidade de armazenamento dela é bem baixa, pois a função dela é somente armazenar os dados que estão sendo processados **naquele momento pela CPU**.

Arquitetura de Computadores

- **Memórias**

Em seguida temos as **memórias cache**.

Essas são mais rápidas do que a RAM, porém, um pouco mais lenta que os registradores.

Elas se localizam juntos da CPU e são utilizadas por algumas estratégias com objetivo de antecipar a necessidade de utilizar um dado e, então, armazená-lo nessas memórias para evitar algumas leituras na memória RAM.

Essas memórias possuem níveis, nível 1, nível 2 e nível 3 e daí seus nomes, l1, l2 e l3. Cada nível é basicamente a localização delas em relação a CPU, tornando-as mais rápidas ou menos rápidas.

Arquitetura de Computadores

- **Memórias**

Em seguida temos a **memória principal** que normalmente é a nossa memória RAM, com 4GB ou 8GB nos computadores populares atuais.

Quanto maior ela for, maior será a área de trabalho que ela proverá aos usuários.

Arquitetura de Computadores

- **Memórias**

Por último temos as memórias **secundárias**, que acabamos de ver os motivos delas serem mais lentas, essas apresentam uma capacidade muito maior de armazenamento e um custo muito inferior em relação aos outros tipos de memória, mas principalmente, **sua grande diferença é que ela armazena os dados de maneira não volátil, então você não perderá seus dados ao desligá-las.**

Arquitetura de Computadores

- Sistema Operacional

Toda essa arquitetura nos parece um pouco complexa de se entender, porém, conseguimos utilizar o computador facilmente sem compreendê-la.

Isso é possível graças aos **sistemas operacionais**, que fazem essa mediação entre a arquitetura, hardware, software e nós usuários, garantindo que tudo funcione da maneira **correta e harmoniosa**.

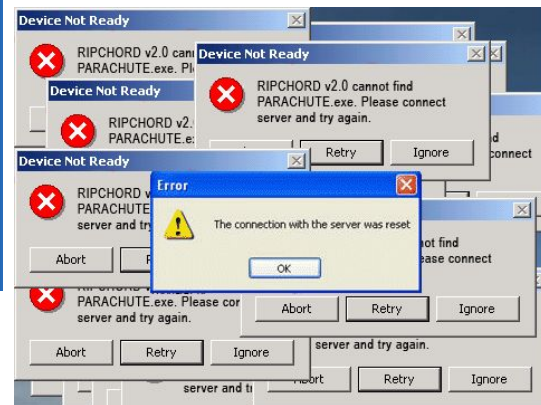
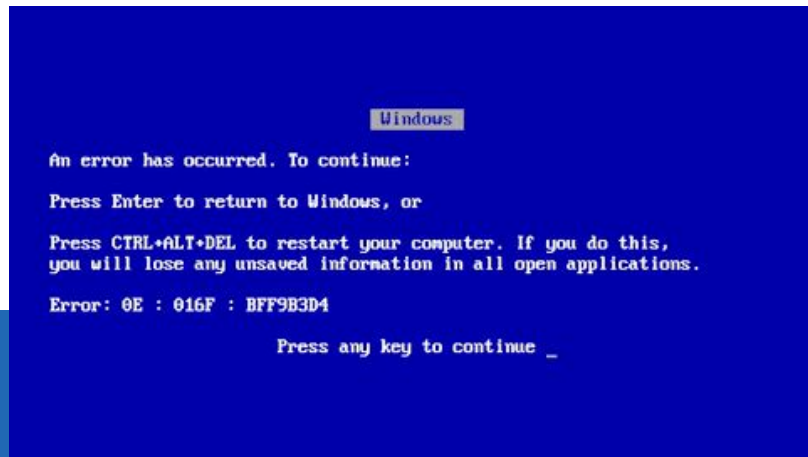
Arquitetura de Computadores

- Sistema Operacional



Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (0% complete)

If you'd like to know more, you can search online later for this error: HAL_INITIALIZATION_FAILED



Arquitetura de Computadores

- Sistema Operacional

Graças aos SOs, foi possível a criação do conceito de **PC** - Personal Computer, ou computadores pessoais, pois permitiu que pessoas comuns, sem conhecimentos técnicos de computação, pudessem utilizar essas máquinas em seu dia-a-dia para diversas tarefas.

Tudo isso graças ao alto nível de abstração com que permitem que cada pessoa usuária interaja com seu hardware.

Arquitetura de Computadores

- Sistema Operacional

Os primeiros sistemas operacionais não possuíam interface gráfica e tudo era realizado através de linhas de comandos.

Hoje conseguimos fazer diversas atividades através de telas **intuitivas** e fáceis de se interagir e, não só isso, desenvolvemos aplicações que exploram essas interfaces, por exemplo, quando criamos nossos frontends, estamos criando uma aplicação para o usuário interagir utilizando a **interface disponibilizada e gerenciada pelo SO**.

Arquitetura de Computadores

- Sistema Operacional

O sistema operacional atua como um **gerenciador** ou administrador de recursos, e está o tempo inteiro gerenciando diversas atividades em seu computador.

Ele controla as **filas de processos** para serem carregados para a **memória principal** e de lá para serem **processados**, monitora os diversos dispositivos do computador, como mouse, teclado ou um pendrive conectado, atualiza constantemente os dados exibidos no monitor e tudo isso sem deixar de atualizar o relógio na bandeja a cada minuto.

Vamos ver um pouco mais sobre esse gerenciamento que o SO faz.

Arquitetura de Computadores

- Sistema Operacional

Um **processo** basicamente é um programa que é colocado em execução: um comando que executamos no terminal, um software que abrimos, um código que rodamos, enfim, tudo isso são processos que podemos executar através da interface do SO.

Não confunda o **processo com o programa**, por exemplo, se clicarmos várias vezes no ícone do navegador serão executados vários processos daquele mesmo programa.

Arquitetura de Computadores

- Sistema Operacional

De modo geral, podemos dividir os processos em dois tipos, de acordo com a forma em que eles são executados.

Existem processos que rodam em **primeiro plano**, ou seja, ao executarmos, ele nos é exibido ou em uma nova janela ou em nosso terminal; e há também processos que executam em **segundo plano**, que chamamos de execução em **background**, por mais que não estejamos vendo uma interface ou o output direto dele, ele está sendo executado pelo computador.

Durante a execução do sistema operacional, diversos processos em background são iniciados.

Arquitetura de Computadores

- **Sistema Operacional**

Cada processo possui uma estrutura básica possuindo:

- Um `process_id` identificador único;
- Um endereço relativo para o SO localizá-lo, esse endereço não é literalmente o endereço na memória. Esse endereço permite o SO saber onde aquele processo se inicia e termina.

Nesse endereço é armazenado o texto do programa, ou seja, o código com as instruções; os dados, como variáveis; e a pilha de execução, quais rotinas já foram chamadas e qual será a próxima;

Arquitetura de Computadores

- **Sistema Operacional**

Cada processo possui uma estrutura básica possuindo:

- Sua instrução, ou seja, qual foi a instrução que iniciou aquele processo;
- Os contextos de Hardware e Software, que são basicamente os dados que são utilizados para aquele processos, por exemplo, qual instrução está nos registradores (hardware) ou os valores atuais das variáveis (software).

Sistema Operacional

Comando Top

```
Processes: 346 total, 2 running, 344 sleeping, 1960 threads
Load Avg: 3.49, 3.89, 4.31 CPU usage: 10.77% user, 8.19% sys, 81.3% idle
SharedLibs: 174M resident, 43M data, 15M linkedit.
MemRegions: 271070 total, 2061M resident, 53M private, 587M shared.
PhysMem: 8159M used (3432M wired), 32M unused.
VM: 1705G vsize, 1993M framework vsize, 170533128(64) swapins, 174159159(0) swapouts.
Networks: packets: 91891621/55G in, 51097723/24G out.
Disks: 57564000/1174G read, 40715346/1036G written.

PID    COMMAND    %CPU    TIME    #TH    #WQ    #PORTS    MEM    PURG    CMPRS    PGRP    PPID    STATE
194    WindowServer 19.3    54:14:02 11     5      2911     352M    0B      128M    194     1      sleeping
39007   Slack Helper 11.6    05:46:23 9       1      168      571M    8192B   418M-   39006   39006   sleeping
39011   Slack Helper 8.2     14:46:02 17      1      274      761M    0B      339M-   39006   39006   sleeping
1495    com.docker.h 7.5     51:06:23 15      1      39       2366M    0B      313M    1476    1489   sleeping
51944   top          6.1     00:49.61 1/1     0      31       7240K+   0B      1448K   51944   39116   running
39006   Slack        3.8     03:41:02 29      1      436      95M     0B      47M     39006   1      sleeping
0       kernel_task  3.8     34:01:58 168/4   0      0         183M    0B      0B      0       0      running
25300   Code Helper  2.7     55:56.40 8        1      156      227M    8192B   17M-    25299   25299   sleeping
25303   Code Helper  2.4     56:47.41 24        1      245      134M+   0B      109M    25299   25299   sleeping
95731   Google Chrom 1.9     01:46:52 17        2      280-     223M+   0B      167M-   1243    1243   sleeping
39114   Terminal     1.1     00:33.30 8         1      297      22M-    2004K+   5192K   39114   1      sleeping
1243    Google Chrom 1.0     10:52:09 42        3      1621     349M    0B      169M-   1243    1      sleeping
28564   AppleUserHID 0.9     56:54.70 3         2      67       756K    0B      524K    28564   1      sleeping
1289    Google Chrom 0.7     04:42:19 9         1      156-     83M-    0B      55M     1243    1243   sleeping
50310   Google Chrom 0.4     33:01.64 15        1      263      483M+   0B      87M     1243    1243   sleeping
28565   AppleUserHID 0.4     26:05.56 2         1      65       900K    0B      520K    28565   1      sleeping
1458    mysqld       0.3     02:03:39 39         0      61       361M    0B      353M    1329    1329   sleeping
1488    vpnkit-bridg 0.3     85:25.35 10         0      30       4068K   0B      1288K-   1476    1476   sleeping
25299   Electron     0.3     11:29.20 29         1      426      55M     0B      21M     25299   1      sleeping
1321    mongod       0.2     02:09:21 32         0      56       18M     0B      12M     1321    1      sleeping
43611   Google Chrom 0.2     03:23:34 18         1      19348    340M-   0B      259M    1243    1243   sleeping
16257   core         0.1     67:17.24 21         1      73       16M     0B      12M     16257   1      sleeping
50960   Google Chrom 0.1     00:45.89 16         2      181      139M    0B      71M     1243    1243   sleeping
51601   Google Chrom 0.1     00:26.25 15         2      176      147M    0B      75M     1243    1243   sleeping
19062   Google Chrom 0.1     13:55.46 14         2      143      99M+    0B      92M     1243    1243   sleeping
23536   Google Chrom 0.0     08:56.70 13         1      253      318M    0B      300M-   1243    1243   sleeping
32349   Google Chrom 0.0     07:28.63 15         2      200      119M+   0B      108M    1243    1243   sleeping
1481    com.docker.v 0.0     09:26.19 5          0      22       35M     0B      30M     1476    1476   sleeping
25313   Code Helper  0.0     02:31.53 20         1      86       44M+    0B      41M     25299   25303   sleeping
87800   Google Chrom 0.0     42:06.00 18         2      394      541M    0B      259M    1243    1243   sleeping
43684   Google Chrom 0.0     30:39.77 15         2      258      238M    0B      220M    1243    1243   sleeping
1752    storagekitd  0.0     13:47.52 7          3      92       3888K   0B      3240K   1752    1      sleeping
29193   CommCenter   0.0     00:32.38 7          2      148      7216K   0B      6244K   29193    1      sleeping
```

Arquitetura de Computadores

- Sistema Operacional

Os primeiros sistemas operacionais eram ***monoprogramados***, o que significa que o processamento era reservado ao processo que estava em execução, mesmo que este estivesse aguardando por alguma operação mais lenta ser concluída.

A maioria dos sistemas operacionais hoje são ***multiprogramados***, o que significa que eles são capazes de **compartilhar** o tempo da CPU, compartilhando-o entre as aplicações que estão em execução, como se fosse um rodízio para utilizar tal recurso, resultando em um **melhor aproveitamento dos recursos e numa melhor performance**, já que o tempo de ociosidade da CPU é diminuído.

Arquitetura de Computadores

- Sistema Operacional

O sistema operacional gerencia os espaços disponíveis e ocupados do computador, alocando e localizando processos e dados na memória.

Ao iniciar um processo, ele é alocado em um espaço da memória e, ao encerrá-lo, esse espaço é desalocado novamente.



Arquitetura de Computadores

- Sistema Operacional

O SO divide a memória em partições e então gerencia filas para os processos aguardarem e então serem alocados, garantindo que um processo não invada o espaço de outro.

Ao adicionar um processo à memória, ele é colocado em um endereço lógico que é utilizado pelo SO, esse endereço é então convertido por um componente chamado de MMU (*memory management unit*) nos endereços físicos do hardware da memória.

Arquitetura de Computadores

- Sistema Operacional

Essas partições podem ser **fixas**, ou seja, são definidas na inicialização do sistema (**boot**) ou, nos ambientes mais recentes, são **dinâmicas**, sendo criadas e alteradas durante a execução dos programas.

Uma função que conseguimos entender um pouco melhor a partir desses conceitos, é o **swapping**, vocês já devem ter ouvido falar sobre swap ou swapping.

Arquitetura de Computadores

- Sistema Operacional

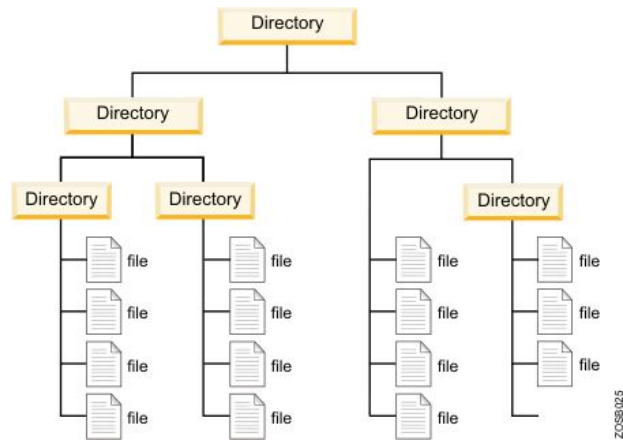
O **swapping** é a transferência de dados entre a memória principal e as secundárias, isso acontece, pois, como vimos, a memória principal tem uma capacidade limitada, e normalmente iremos ter processos demandando mais memória do que o existente.

Para resolver isso, o SO move aquelas partes da memória que não estão sendo utilizadas naquele momento para a **área de swap no disco**, que fica na memória secundária do computador, **um HD**, e quando essa informação voltar a ser requerida, o processo inverso é realizado retornando os dados para a **memória principal**.

Arquitetura de Computadores

- Sistema de Arquivos

A parte que mais interagimos com SO é com seu módulo de **gerenciamento de arquivos**, ao realizar o login na máquina, a primeira coisa que nos deparamos é com o Desktop, que nada mais é, na sua essência, do que uma **representação gráfica do diretório**.



Arquitetura de Computadores

- Sistema de Arquivos

Nossos comandos são entendidos pelo SO que faz toda a gerência de como aquele dado será tratado até chegar em uma sequência de bits no HD.

Um processo pode ler ou escrever arquivos, assim como criar novos ou apagar existentes. **Essas ações são todas realizadas por meio do SO.**

Arquitetura de Computadores

- API SO

Um recurso interessante para conhecermos é a **API** que os sistemas operacionais disponibilizam, quando precisamos de um desses gerenciamentos podemos solicitar direto ao sistema operacional de maneira abstraída e então ele saberá fazer os devidos tratamentos.

Arquitetura de Computadores

- API SO

Ao abstrair essas funções, não importa qual versão do SO ou quais hardwares estamos utilizando, a chamada terá um padrão. Esse padrão também é abstraído pelas linguagens de programação através de uma sintaxe ainda mais simples.

Permitindo assim que facilmente consigamos alocar um endereço de memória para armazenar uma variável ou interagir com o sistema de arquivos, por exemplo.



betrybe.com

Obrigado!