



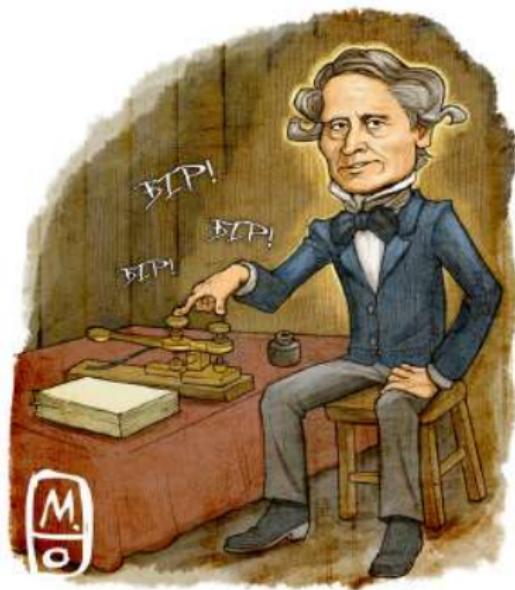
UNIVERSIDADE  
**VILA VELHA**  
ESPIRITO SANTO

UNIX

Prof. Jean-Rémi Bourguet

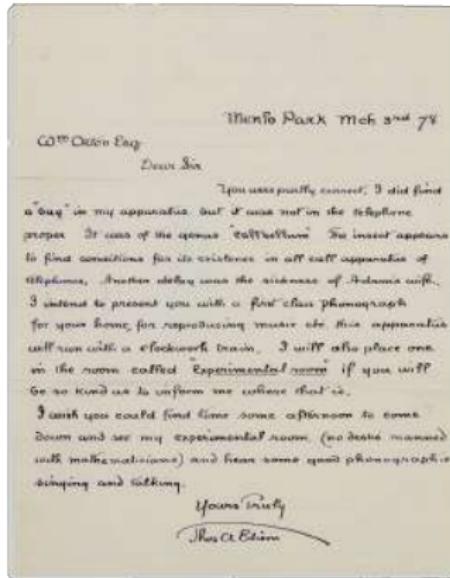
Sistemas Operacionais

- Em 1840, **Morse** patenteou a transmissão de **sinais / sons por fios**.



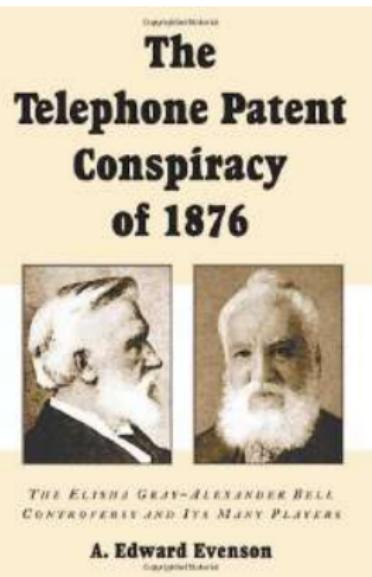
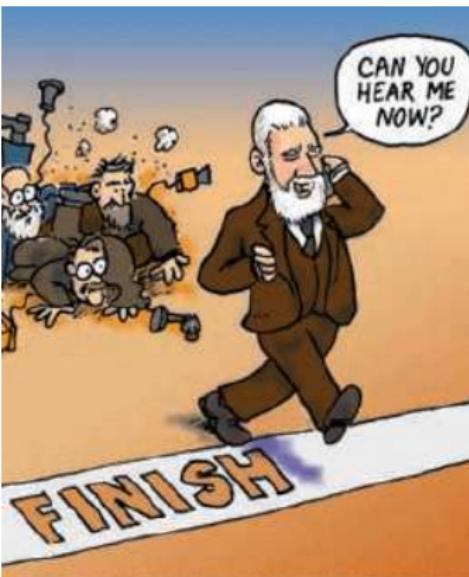
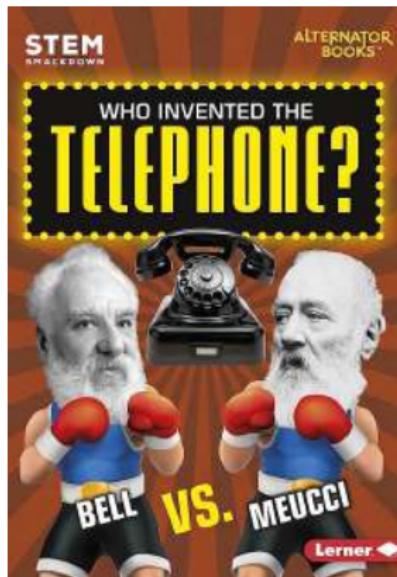
# Bell Labs

- Thomas Edison aprendeu o Morse e construiu telégrafos artesanais.
- Em 1978, encontrou o **primeiro bug**. Em 1879, **patentiou** o telegrafo.



# Bell Labs

- Em 1876, A. G. Bell ganhou a **corrida** pela **invenção do telefone** (voz).

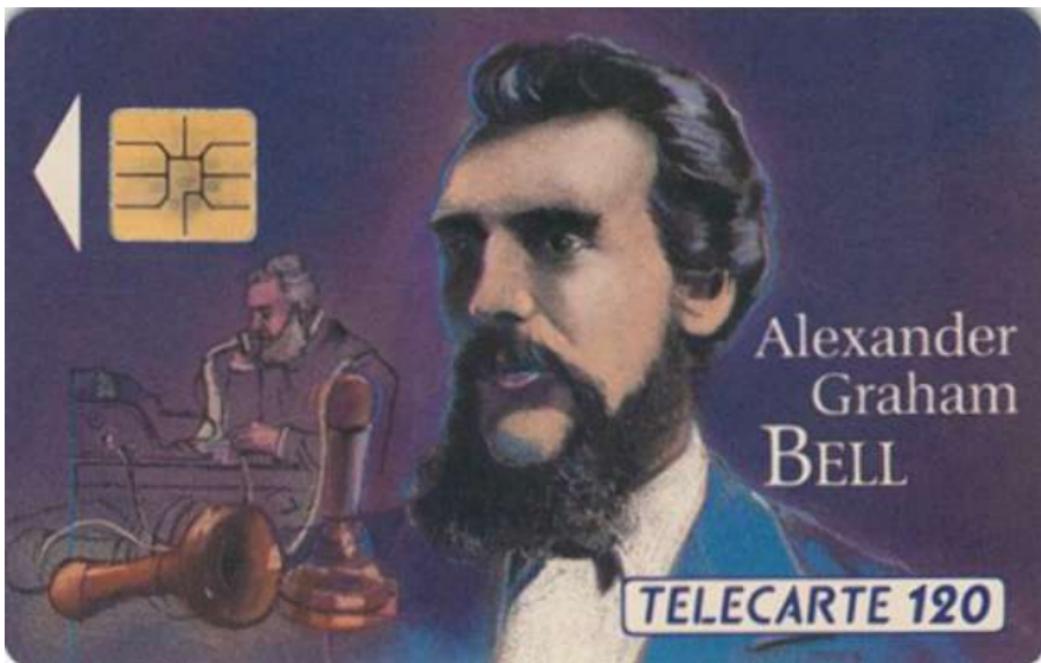


- AT&T (Bell Patent Association) protegeu os direitos de patente.

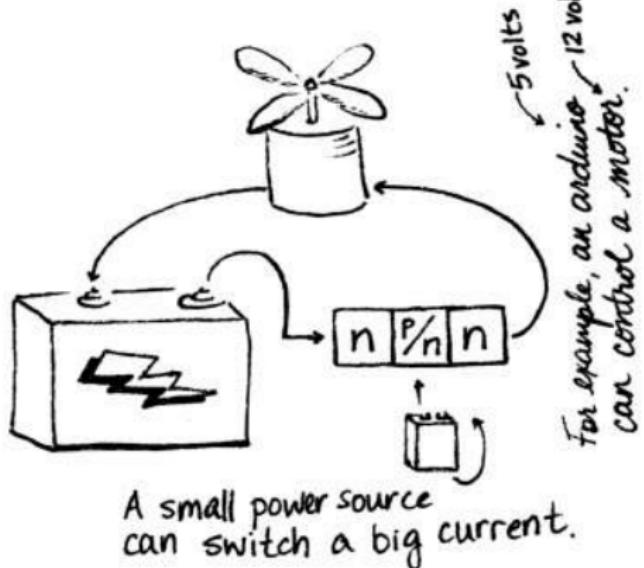


# Bell Labs

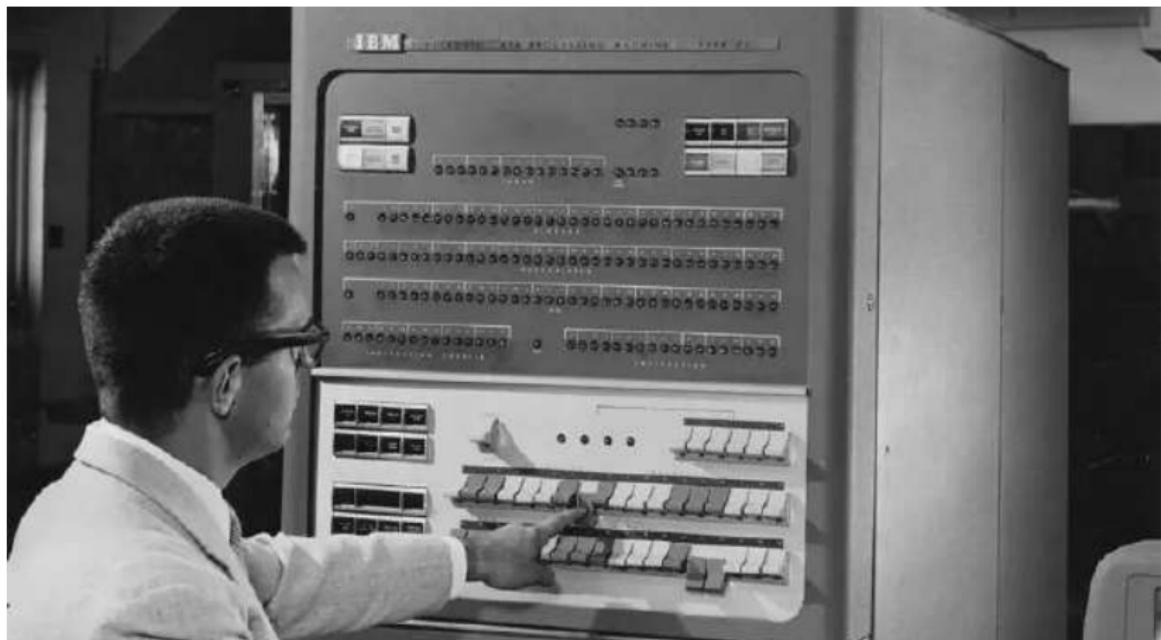
- Em 1880, o governo francês concedeu a Bell o **Prêmio Volta** (50.000 F).
- Financiou o Volta Lab que se tornou o **Bell Lab** da AT&T em 1925.



- Em 1947, Bardeen, Brattain e Shockley inventaram o **transistor**.



- Em 1952, Bell Labs começou a explorar a **comutação eletrônica**.
- Sistemas de **telefonia pública** e de troca de negócios privados.



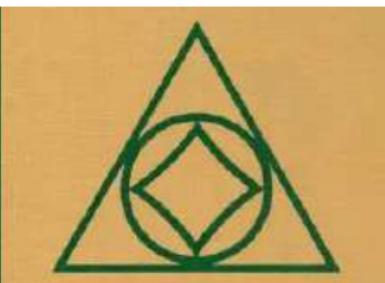
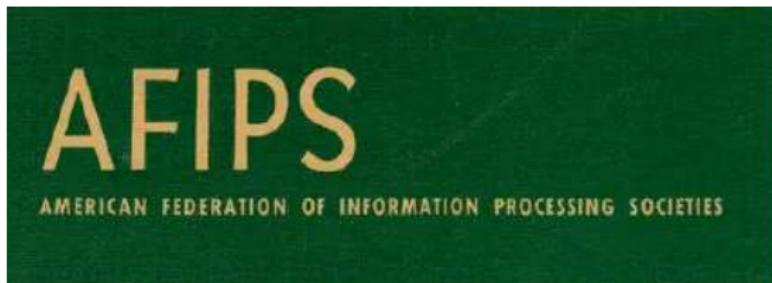
\* Computer System Research department at Bell Labs

- Em 1965, **Ed David Jr** (Bell Labs) foi co-autor de **Robert Fano** (MIT).

## SOME THOUGHTS ABOUT THE SOCIAL IMPLICATIONS OF ACCESSIBLE COMPUTING\*



E. E. David, Jr.  
*Bell Telephone Laboratories, Inc.*  
Murray Hill, New Jersey  
and  
R. M. Fano  
*Massachusetts Institute of Technology*  
Cambridge, Massachusetts



- Tecnologia de **cartões perfurados Hollerith** (patentes de mecanografia).
- Thomas J. **Watson** é nomeado chefe da C-T-R e a **renomeia como IBM**.



International Time Recording  
Company (1888)



Computing Scale  
Company (1891)



Computing-Tabulating-  
Recording Company (1911)



International Business  
Machines (1924)

# IBM

1947

# IBM

1956

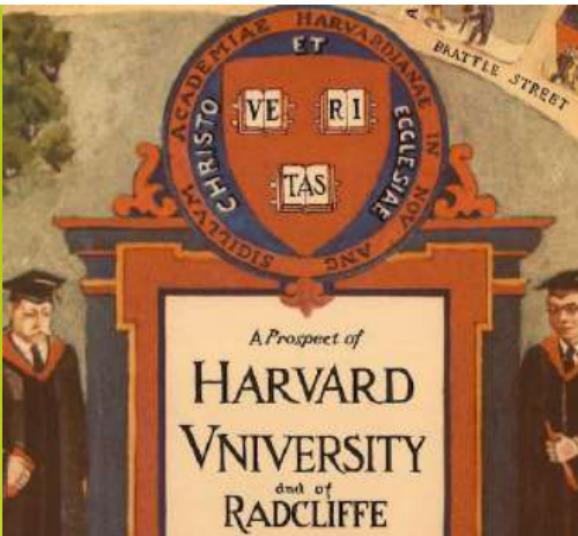
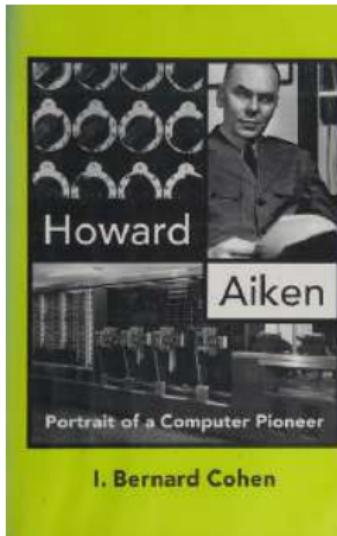
1972

- Em 1937, **governo** dos EUA usou equipamentos de **tabulação da IBM**.
- Registros de **26 milhões** de beneficiários da Lei da Previdência Social.

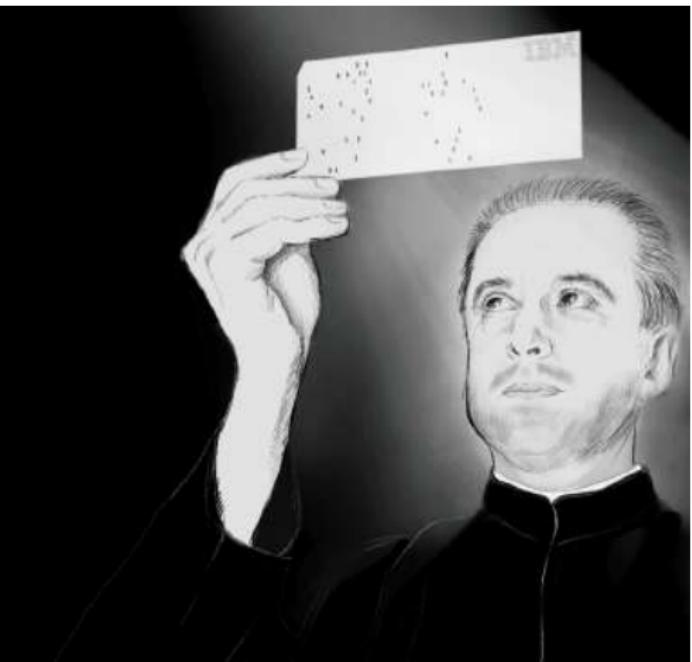


*Automatically obtain*  
**Employer, Employee Records**  
*for the*  
**SOCIAL SECURITY PROGRAM**

- Harvard Mark I primeira calculadora eletro-mecânica de **larga escala**.
- Em 1930, começou com um aluno e construído em **1944 com IBM**.



- Thomas J. Watson encontra o jesuíta italiano **Roberto Busa** em 1949.
- **Precursor do hypertext** 15 anos antes das pesquisas norte-americanas.



- ▶ **IBM 700/7000:** série de computadores de grande tamanho (mainframe).
- ▶ 700s usavam **tubos de vácuo** (50's) e os 7000s **transistores** (60's).



► Em 1892, Edison General Electric e Thomson-Houston Electric **fusionam**.



1892-1900



1900-1909



1909-PRESENT

## This Room Is Equipped With Edison Electric Light.

Do not attempt to light with  
match. Simply turn key  
on wall by the door.

The use of Electricity for lighting is in no way harmful  
to health, nor does it affect the soundness of sleep.

# General Electric

- No início dos 60's, a GE era a maior usuária de **mainframes IBM**.
- Decidiu produzir suas próprias máquinas **reduz os custos**.



- A série GE 600 eram de **segunda geração com transistores**.
- Em 1964, IBM considerou uma **competição severa**.



# John Backus

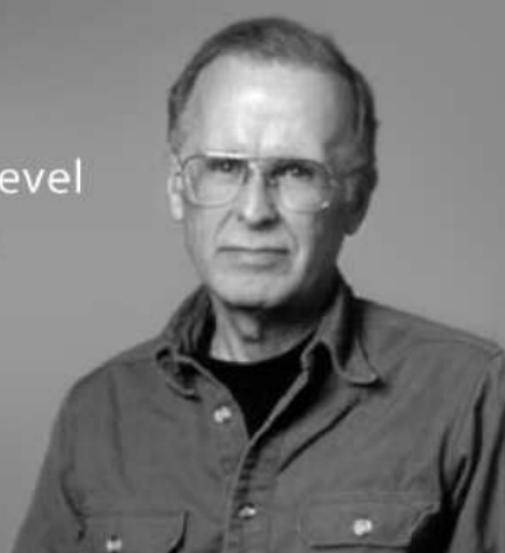
IBM

- Em 1957, John Backus criou a 1<sup>a</sup> linguagem de prog de alto nível.

# JOHN BACKUS

Profoundly influenced high-level  
programming system design  
through work on FORTRAN.

acm A.M.  
**TURING** AWARD  
1977

A black and white portrait photograph of John Backus, an elderly man with glasses and short hair, wearing a button-down shirt, looking slightly to the right of the camera.

- **FORTAN (FORMula TRANslator)** foi desenvolvido para **IBM 704**.



<http://hackaday.com/2015/10/26/this-is-not-your-fathers-fortran>

## 23 An Experimental Time-Sharing System (1962)

Publisher: MIT Press

Cite This

PDF

By Fernando Corbató , Marjorie Merwin Daggett , Robert C. Daley

Book Chapter

is part of: Ideas That Created the Future: Classic Papers of Computer Science

Editor(s): Harry R. Lewis Show More

22

Downloads



### Abstract

### Chapters & Sections

» Ideas That Created the Future: Classic Papers of Computer Science

» Preface

» Introduction: The Roots and Growth of Computer Science

### Chapter Abstract:

By the mid-1950s, researchers had enough experience with building and using electronic computers to begin imagining different ways computers might evolve so that more people could use them to solve more problems. In a 1954 summer school at MIT, a remarkable exchange occurred between Grace Hopper and John Backus, who would soon thereafter develop the Fortran programming language. "Dr. Grace Hopper raised the possibility of using several small computers in parallel. The greatest demand was for small machines. ... She foresaw a mass produced small machine, delivered with a compiler and library appropriate to the customer's needs. Mr. J. W. Backus disagreed with this philosophy on the grounds of computer speed; since increased speed costs little more, a large computer is cheaper to use than a small one.... John Backus said that by time sharing, a big computer could be used as several small ones; there would need to be a reading station for each user" (Adams et al., 1954, pp. 16-1-16-2).

*"Um grande computador poderia ser usado como vários pequenos.  
Seria necessário haver uma estação de leitura para cada usuário".*

**John Backus, Summer school at MIT, 1954.**



<https://ieeexplore.ieee.org/document/9357648>

# Time Sharing no MIT

- Em 1959, **Christopher Strachey** imaginava depurar um programa...  
...em teletype enquanto **outro roda** no computador ao **mesmo tempo**.
- Neste evento, ele passou o conceito para **J.C.R. Licklider** do **MIT**.



# SEMANTIC SCHOLAR

Corpus ID: 5144680

**Time sharing in large, fast computers**

C. Strachey - Published in IFIP Congress 1959 - Computer Science

No Paper Link Available | Save to Library | Create Alert | Cite

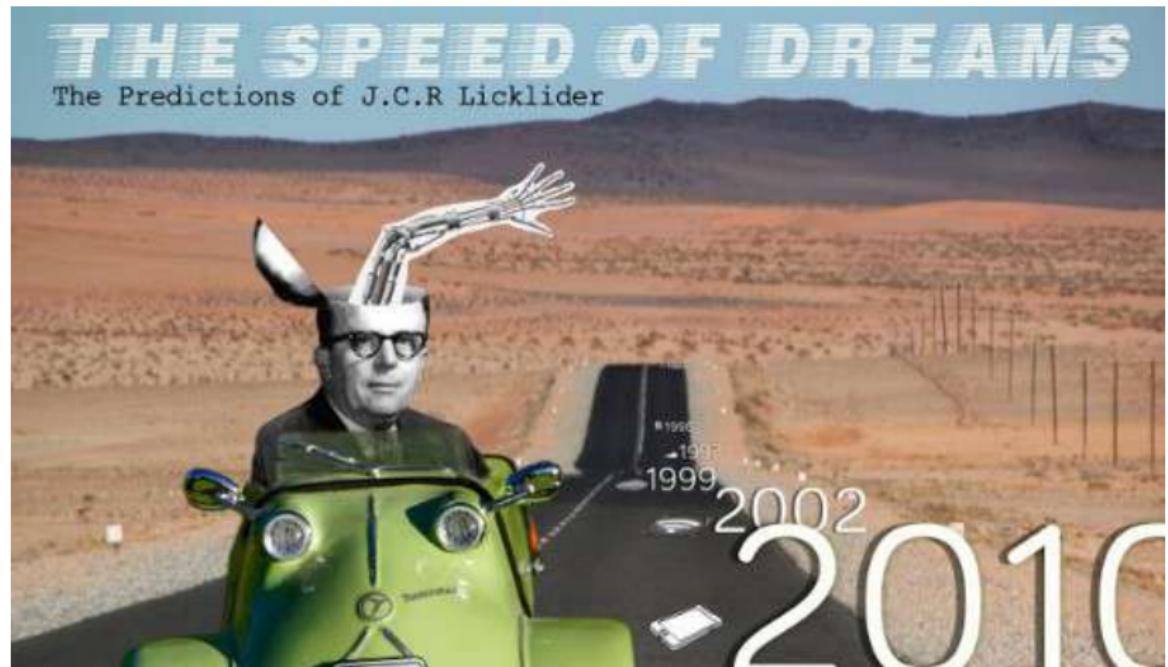
Share This Paper 

<b>116 Citations</b>
Highly Influential Citations 4
Background Citations 31
Methodist Citations 5
<a href="#">View All</a>

 <https://api.semanticscholar.org/CorpusID:5144680>

# Joseph Carl Robnett Licklider

Harvard / MIT



🔗 <https://medium.com/@louis.cadier/the-speed-of-dreams-the-predictions-of-j-c-r-licklider-5d8cb84fbbfe>

# Joseph Carl Robnett Licklider

Harvard / MIT

- Em 1957, deixou o MIT para ser **vice** da Bolt Beranek & Newman.
- Em 1962, desenvolveram **BBN Time-Sharing System** (demo pública).



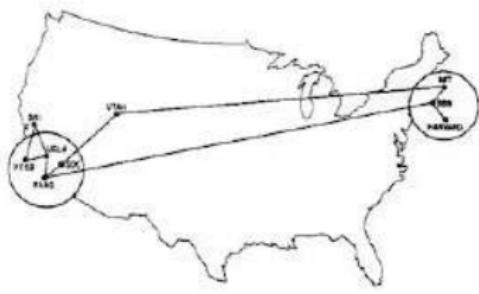
# Joseph Carl Robnett Licklider

Harvard / MIT

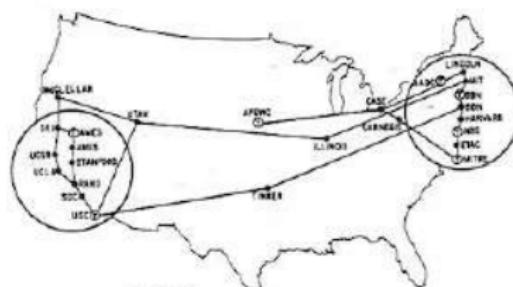
- Em 1962, foi nomeado **chefe de programas** da ARPA.
- Por fim, sua visão levou à **ARPANet**, a **precursora da Internet** atual.



1969



1970



1972



1977

# Joseph Carl Robnett Licklider

Harvard / MIT

- ▶ Voltou ao **MIT como professor** de engenharia elétrica em 1968.
- ▶ Ele atuou simultaneamente como diretor do **Projeto MAC** até 1971.



🔗 <https://mac50.csail.mit.edu/index.html>

\* Em 2003, CSAIL reuniu os elementos diversificados do Projeto MAC.

# John McCarthy

Stanford University / Dartmouth / MIT

- ▶ Ajudou na **criação do Projeto MAC** no MIT quando trabalhou lá.
- ▶ Ajudou a criar o **Stanford AI Laboratory** rival amigável do MAC.



# John McCarthy

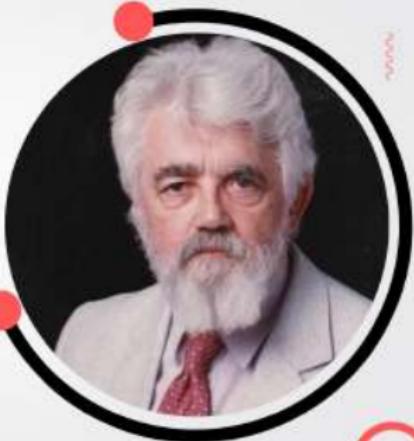
Stanford University / Dartmouth / MIT

*"The Internet would not have happened nearly as soon as it did except for the fact that John initiated the development of time-sharing systems. We keep inventing new names for time-sharing. It came to be called servers ... Now we call it cloud computing. That is still just time-sharing. John started it.*

Woo, Elaine

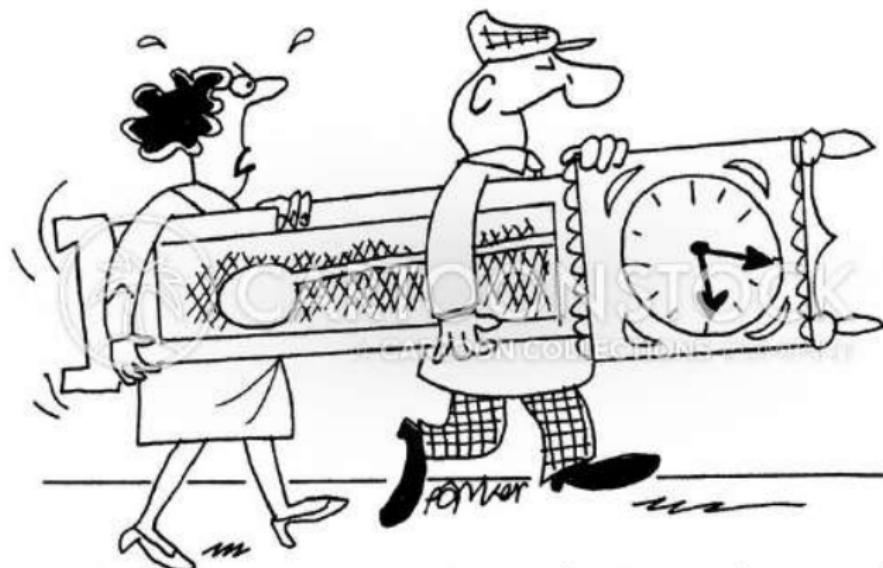
MEET THE FATHER OF CLOUD COMPUTING

# JOHN MCCARTHY

A circular portrait of John McCarthy, an elderly man with white hair and a beard, wearing a suit and tie. He is positioned in the center of a white background with a black circular border. Red decorative dots are scattered around the border.

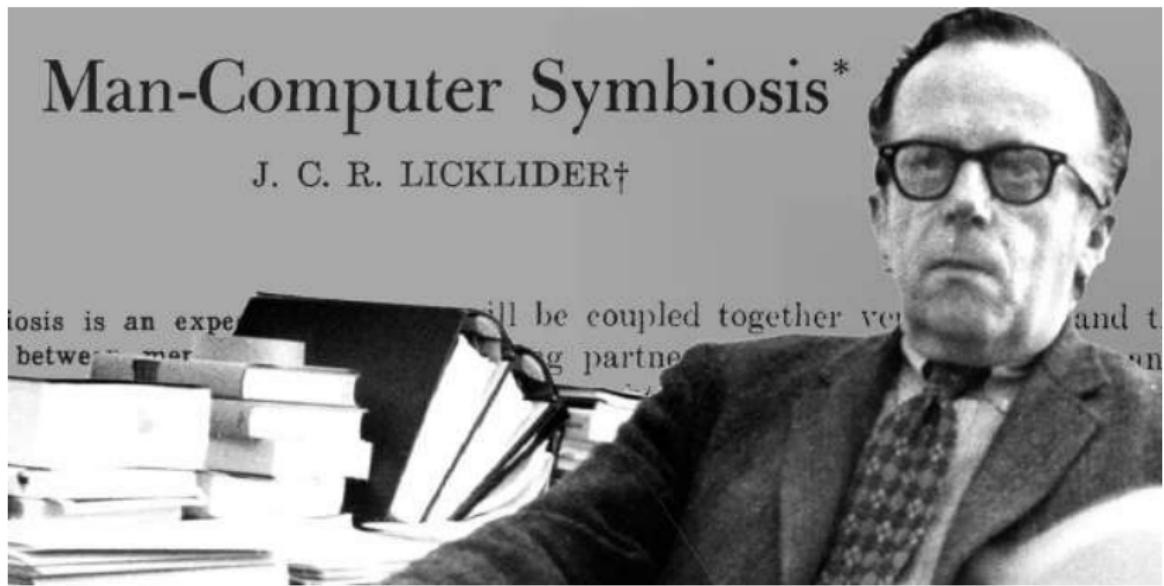
# Time Sharing no MIT

- ▶ Time-sharing é o **compartilhamento** de um **recurso computacional**.
- ▶ **Vários usuários** ao **mesmo tempo**: **multi-programação** e **multi-tarefa**.



"Well, it's not MY idea of time sharing!"

- Em 1959, McCarthy escreveu um **memorando** sobre isso no MIT.
- Em 1960, **Licklider** abraçou a ideia de timesharing também.



## Man-Computer Symbiosis\*

J. C. R. LICKLIDER†

iosis is an exper... will be coupled together very closely between man and machine, and t...  
...n

- 🔗 <https://web.stanford.edu/~learnest/nets/timesharing.htm>
- 🔗 <https://groups.csail.mit.edu/medg/people/psz/Licklider.html>

- Em 1961, alguns comandos tornaram-se operacionais **no IBM 709**.
- **Fernando J. Corbató** demonstrou no MIT um **sistema experimental**.



- ▶ Primeiro uso de senhas para proteger o acesso a arquivos.
- ▶ Também conhecido através da **lei de Corbato**.



## Corbató's Law:

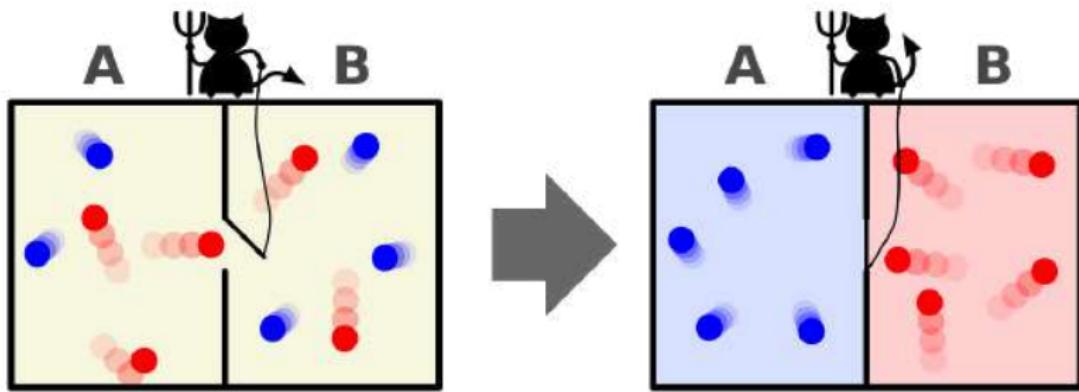
The number of lines of code a programmer can write in a fixed period of time is the same, independent of the language used.



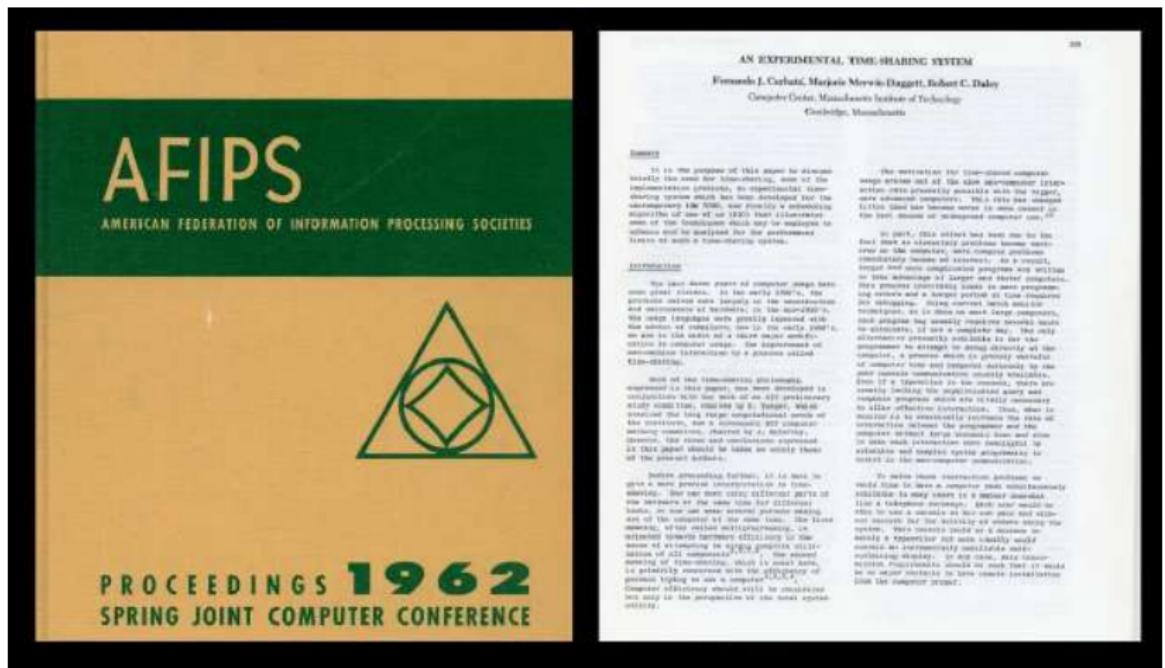
- ▶ **Pioneiro** no desenvolvimento de OS de **compartilhamento de tempo**.
- ▶ **CTSS** (**C**ompatible **T**ime-**S**haring **S**ystem) um dos primeiros no **MIT**.



- ▶ Um dos **termos técnicos herdados** pelo CTSS é **daemon**.
- ▶ **Maxwell's daemon**: ser imaginário de uma expe de termodinâmica
- ▶ Derivados da BSD têm adotado um **demônio como mascote**.



- ▶ Descrito em um artigo da **Spring Joint Computer Conference** 1962.
  - ▶ **Influenciou** o design de outros sistemas de time-sharing.



# Projeto MAC

- Em **1962**, J.C.R. **Licklider** (MIT) propôs o Projeto **MAC**.
- **DARPA** apoiou financiando com uma doação inicial de **2 milhões \$**.
- **MAC** para **Multiple Access Computers** e/ou **Man And Computers!**



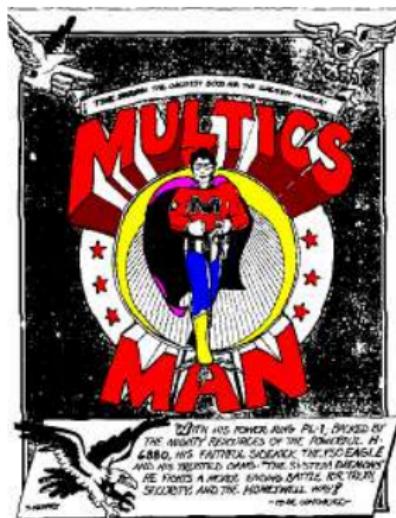
# Projeto MAC

- ▶ Uma das primeiras contribuições foi **fornecer hardware** para CTSS.
- ▶ Em 6 meses, **200 users** acessaram **CTSS em 10 departamentos** do MIT.



# MULTICS

- Em 1969, o **Projeto MAC**, **Bell Labs** e **GE** desenvolveram o MULTICS.
- **MULTICS: MULT**plexed **I**nformation & **C**omputing **S.**

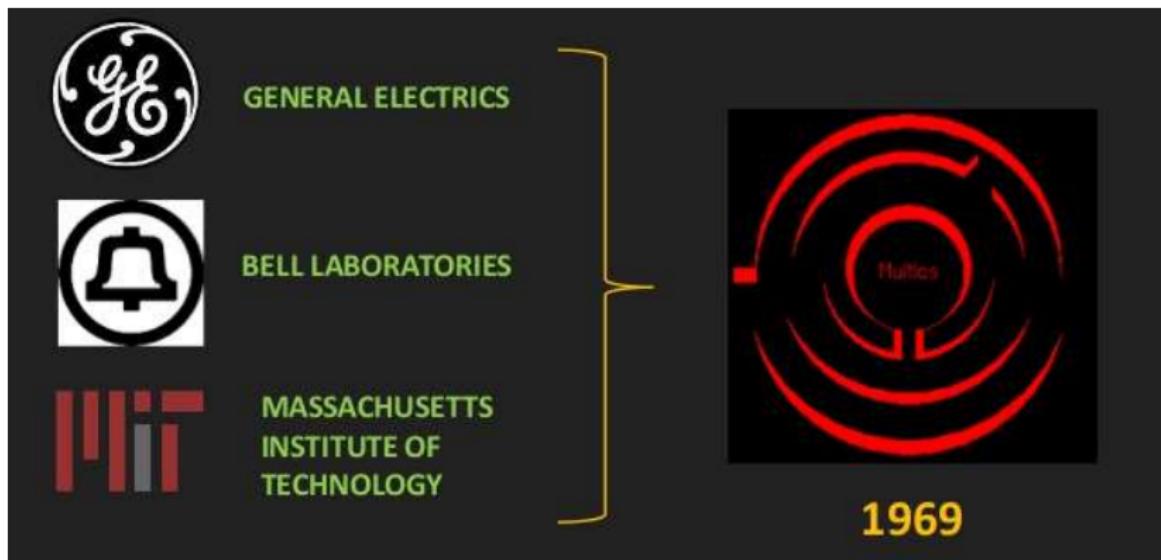


- David Jr (Bell Labs) considerado por Corby **líder da facção pró-MULTICS.**



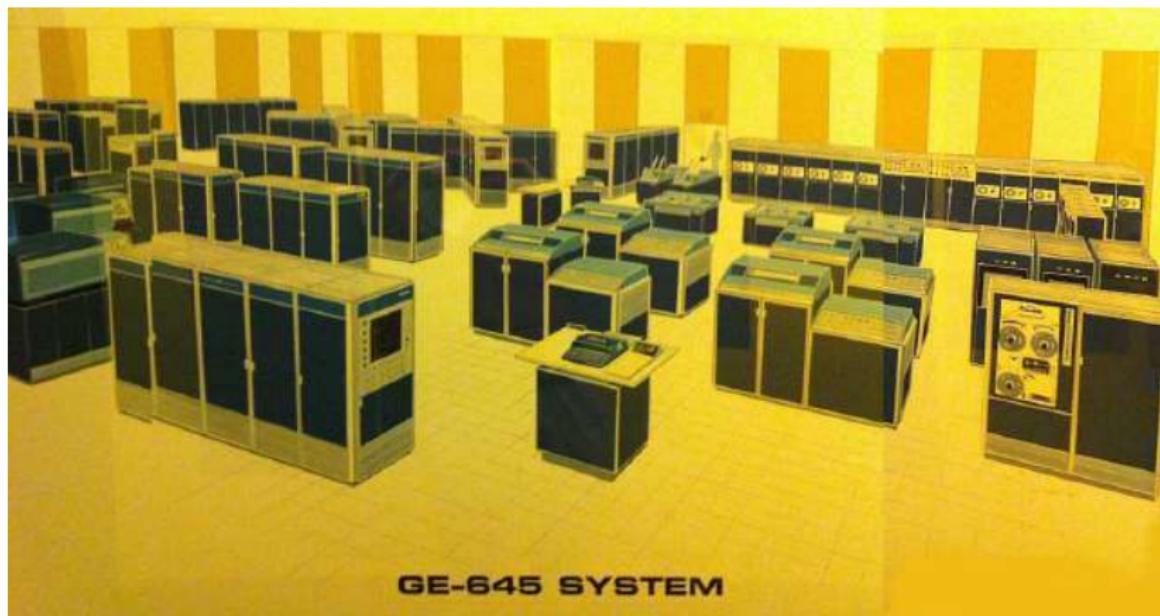
# MULTICS

- Bell Labs (AT&T) comprou um GE-645 da General Electric em 1965.
- Se juntou à equipe de desenvolvimento do MULTICS, no MIT.



# MULTICS

- Em 1969, o **MULTICS** foi disponibilizado para a **comercialização**.
- **Várias organizações** (GE, Ford, US Air Force) **utilizaram** MULTICS.



\* O desenvolvimento do MULTICS foi cancelado somente em 1985!

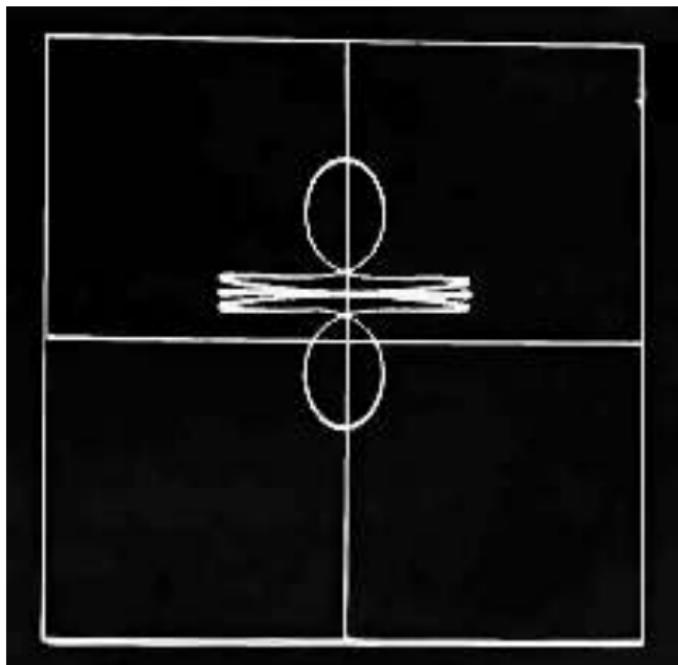
- Bell Labs emprestou alguns programadores para o MIT!



# Ken Thompson

University of California, Berkeley

- Em 1966, Ken Thompson foi contratado pela Bell Labs.
- Em 1969, Ken Thompson desenvolveu o videogame Space Travel.



- ▶ Também, já tinha reescrito o **espetacular editor QED** para **CTSS**.
- ▶ Thompson reescreveu **QED** em **BCPL** para **MULTICS**.

The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "cornel@silentwing: /home/cornel". The window contains the man page for the "ed" command:

```
NAME
    ed, red - text editor

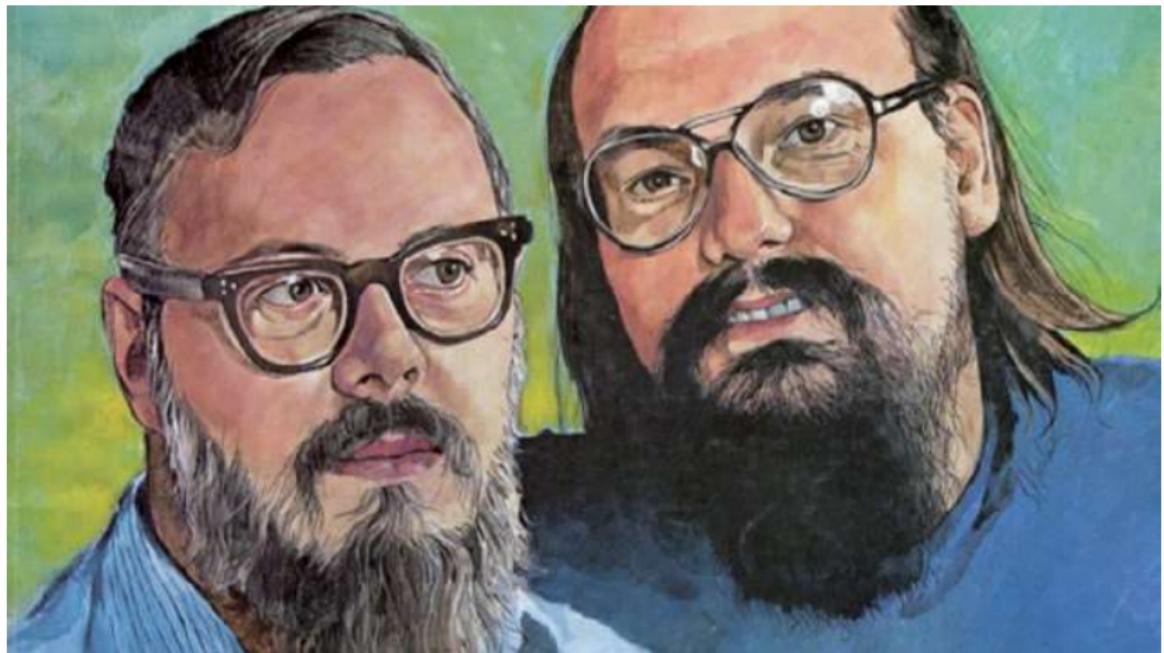
SYNOPSIS
    ed [-] [-Gs] [-p string] [file]
    red [-] [-Gs] [-p string] [file]

DESCRIPTION
    ed is a line-oriented text editor. It is used to create, display, modify and otherwise manipulate text files. red is a restricted ed; it can only edit files in the current directory and cannot execute shell commands.

    Editing is done in two distinct modes: command and input. When first invoked, ed is in command mode. In this mode commands are read from
```

Manual page ed(1) line 1/573 3%

- Thompson e Ritchie desenvolveram um projeto pessoal: UNICS.



- UNICS: UNiplexed Information and Computing Service.
- Trocadilho em relação a MULTICS: UNICS é MULTICS modesto!



- Depois, em 1970, o nome foi mudado de **UNICS** para **UNIX**.

My  
UNIX  
will go on



<https://www.youtube.com/watch?v=g3jOJfr0knA>

- ▶ Era uma tentativa de fazer um **OS simples, versátil e moderno.**
- ▶ Mantem-se idéias de **time-sharing/portabilidade** entre computadores.



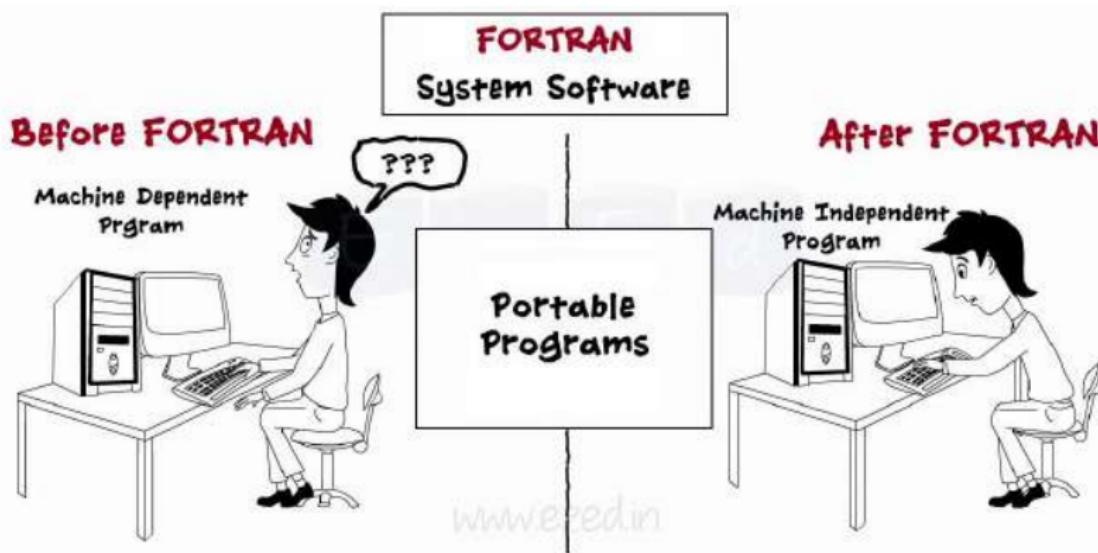
- ▶ Toda a **teoria** sobre o **filesystem** (para utilizar corretamente os **discos**).
- ▶ Toda a **teoria** sobre o **kernel** (intermediação processos/hardware).



- Trabalhando com os seguintes blocos:
  - sistema operacional
  - ambiente shell
  - editor de texto
  - compilação (sistema e programas).
- A **linguagem de programação** utilizada foi a **Assembly**.



- Thompson queria passar UNIX pra uma linguagem de alto nível.
- A primeira tentativa foi Fortran, em 1971.



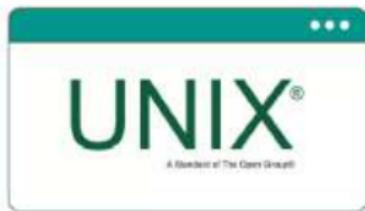
www.wefedin

- Depois, ele escreveu uma linguagem de programação simples: **B**.
- **B** era simplificação de **Basic Combined Programming Language** de 1967.



- ▶ Primeira versão do **UNIX** interna ao MIT foi lançada em **1971** em **B.**
- ▶ Mas havia uma **lentidão da linguagem** (por ser de alto nível).

Slowest things on earth:



# Dennis Ritchie

Harvard

- Em 1967, **Ritchie** começou a trabalhar no **Bell Labs**.
- Nunca enviou uma **cópia encadernada** de sua dissertação de PhD!

156

$\begin{aligned} g^{(y+1)}(x) &< r(y+1) \cdot (\ln(x)+1), x) \quad \text{for } x \geq r_2(y) \\ &\leq r(y+1) \cdot (\sqrt{r^{-1}(x)} + 1), x) \\ &\leq r(r^{-1}(x), x) \quad \text{for } x \geq r_2(y+1) \\ &= r(r^{-1}(x))(x) \end{aligned}$

The third line follows since it is easily shown that  $(y+1) \cdot \sqrt{r^{-1}(x)+1} < r^{-1}(x)$  when  $x \geq (y+2)^2$ ; but since  $r$  is increasing,  $r_2(y+1) < r((y+2)^2) \geq (y+2)^2$ . Then since  $rr^{-1}(x) \leq x$  if  $x \geq r(0)$ , and since for all  $y$   $r_2(y) \geq r(0)$ ,  $r(r^{-1}(x))(x) < h(x)$  by the assumption on  $r$ . Therefore  $g^{(y)}(x) < h(x)$  for  $x \geq r_2(y)$ ; that is,  $g \in_{F_2} h$ . Lemma (14.13) is proved.

(14.14) Theorem. Say  $1 \leq B < \beta < \omega^{\mathbb{N}}$ . Then there is a family  $B$  of classes of functions such that

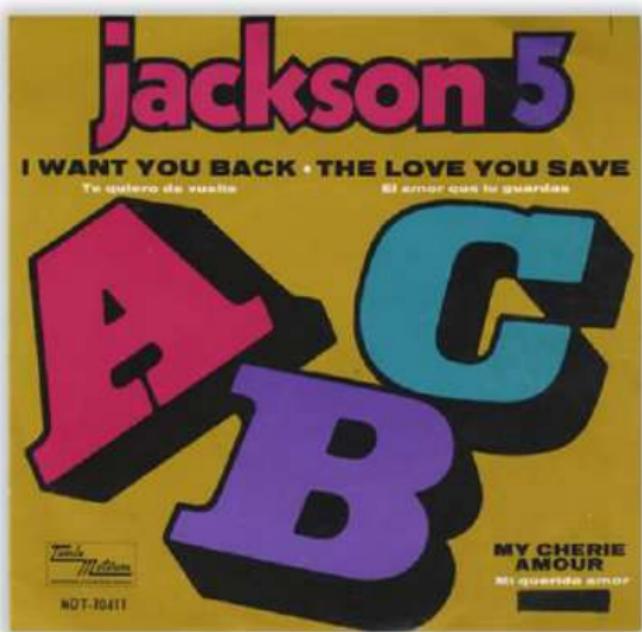
- If  $D \in B$ ,  $Z_D = D = Z_{D'}$
- $B$  has a dense, linear ordering under set inclusion;
- If  $D \in B$ ,  $D$  is fully closed and closed under limited recursion;
- If  $E_1, E_2 \in B$  and  $E_1 \subset E_2$ ,  $E_2$  contains a universal function for  $E_1$ .

Proof. By theorem (14.13), choose an honest, increasing function



<https://www.cs.princeton.edu/~bwk/dmr/doceng22.pdf>

- Ritchie **adicionou funcionalidades** ao B, chamando-se **NB** (New B).
- Ele fiz um **compilador** para o NB e **nasceu** a famosa linguagem **C**.



- Em 1973, **Ritchie fechou** um **acordo** de cooperação **com Thompson**.
- Novamente, **refizeram todo o sistema em** linguagem **C!**

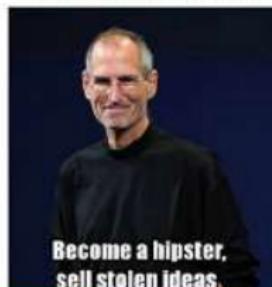


<https://www.youtube.com/watch?v=yY6YY81P3lE>  
 [https://www.youtube.com/watch?v=HVoqoH\\_RMr8](https://www.youtube.com/watch?v=HVoqoH_RMr8)

# Dennis Ritchie

Harvard

Steve Jobs  
(b. 24 Feb 1955)



Become a hipster,  
sell stolen ideas

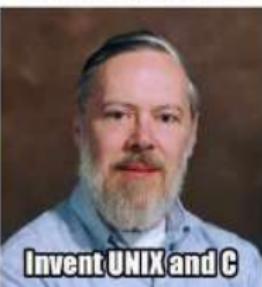
STANCAZ.COM

DIED 05 OCT 2011

Without Steve Jobs, we would have: Without Dennis Ritchie, we would have:

- No iProducts
- No over-expensive laptops

Dennis Ritchie  
(b. 09 Sep 1941)



Invent UNIX and C

DIED 12 OCT 2011

- No Windows
- No UNIX
- No C
- No programs
- A large setback in computing
- No generic text languages
- We would all read in Binary

Praised by media as Jesus  
of computing

Ignored

They died in the same year, but it seems only few notice  
the death of Dennis Ritchie compared to Steve Jobs

XENIX • VERSION 7 • SYSTEM III • SYSTEM V • 4.2BSD • WORK-ALIKES

# UNIX/WORLD

THE MAGAZINE FOR MULTIUSER, MULTITASKING SYSTEMS

NOVEMBER 1981 \$3 IN U.S.A. £2.20 IN U.K. SILICON VALLEY PUBLISHING

The cover features a man in a suit standing on a rocky surface against a starry space background with planets. A small inset window shows a Zilog System 8000 computer system. Headlines include "REVIEW: Zilog System 8000", "Zilog's Mac Connection", "Is The Unix System Retaliatory?", "Graphics Extensions: GKS, PHIGS, VDI", "Tutorial: Terminal Lines, Cron", and "Record: Sun, AT&T Detente, New Fortune Products". A barcode is at the bottom left, and the title "BILL GATES ON THE FUTURE OF XENIX" is at the bottom right.