



# **Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

## **Disciplina: Sistemas Operacionais I**

### **Aula 04: Estruturas dos Sistemas Operacionais P1**

Prof. Diogo Branquinho Ramos

[diogo.branquinho@fatec.sp.gov.br](mailto:diogo.branquinho@fatec.sp.gov.br)

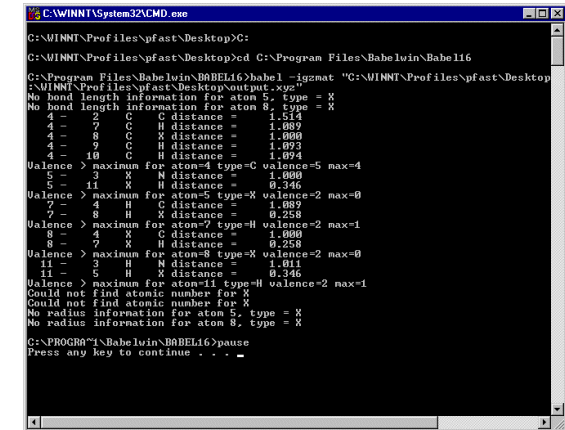
São José dos Campos - SP

# Roteiro

- Interfaces com o usuário
- API
- Chamadas de Sistema
- Parâmetros na Chamada de Sistema
- Monotarefa x Multitarefa

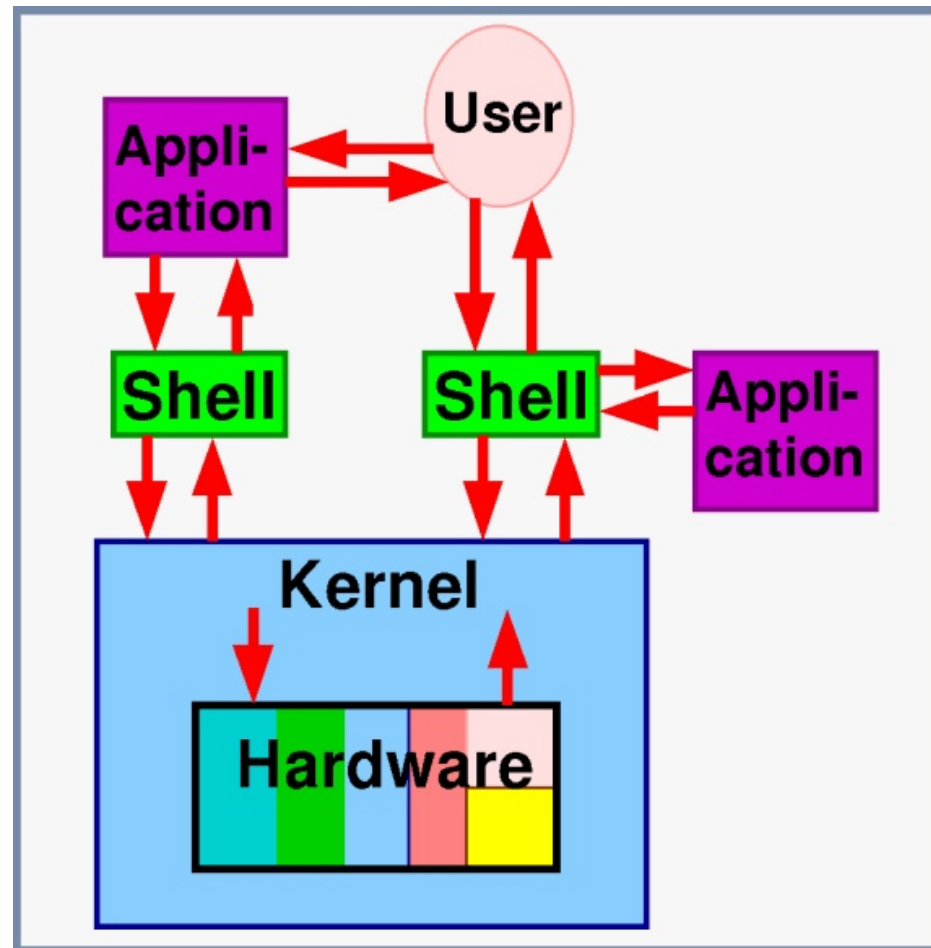
# Interface de comando do SO: CLI

- Interpretador de comandos
  - Interface entre o usuário e o SO.
  - Alguns fabricantes de SOs o incluem no kernel.
    - MS-DOS e UNIX, tratam como um programa especial.
- Shell
  - Programa que lê e interpreta instruções de controle automaticamente.



```
C:\WINNT\System32\CMD.exe
C:\WINNT\Profiles\pfast\Desktop>cd C:\Program Files\Babelwin\Babel16
C:\Program Files\Babelwin\Babel16>babel -igmat "C:\WINNT\Profiles\pfast\Desktop\
WINNT\Profiles\pfast\Desktop\output.xyz"
No bond length information for atom 5, type = X
No bond length information for atom 8, type = X
4 - 2 C C distance = 1.514
4 - 7 C H distance = 1.089
4 - 8 C X distance = 1.000
4 - 9 C H distance = 1.093
4 - 10 C H distance = 1.094
Valence > maximum for atom=4 type=C valence=5 max=4
5 - 3 N N distance = 1.000
5 - 11 X H distance = 0.346
Valence > maximum for atom=5 type=X valence=2 max=0
7 - 4 H C distance = 1.089
7 - 8 H X distance = 0.258
Valence > maximum for atom=7 type=H valence=2 max=1
8 - 4 X C distance = 1.000
8 - 7 X H distance = 0.258
Valence > maximum for atom=8 type=X valence=2 max=0
11 - 2 H N distance = 1.011
11 - 5 H X distance = 0.346
Valence > maximum for atom=11 type=H valence=2 max=1
Could not find atomic number for X
No radius information for atom 5, type = X
No radius information for atom 8, type = X
C:\PROGRAM FILES\Babelwin\Babel16>pause
Press any key to continue . . .
```

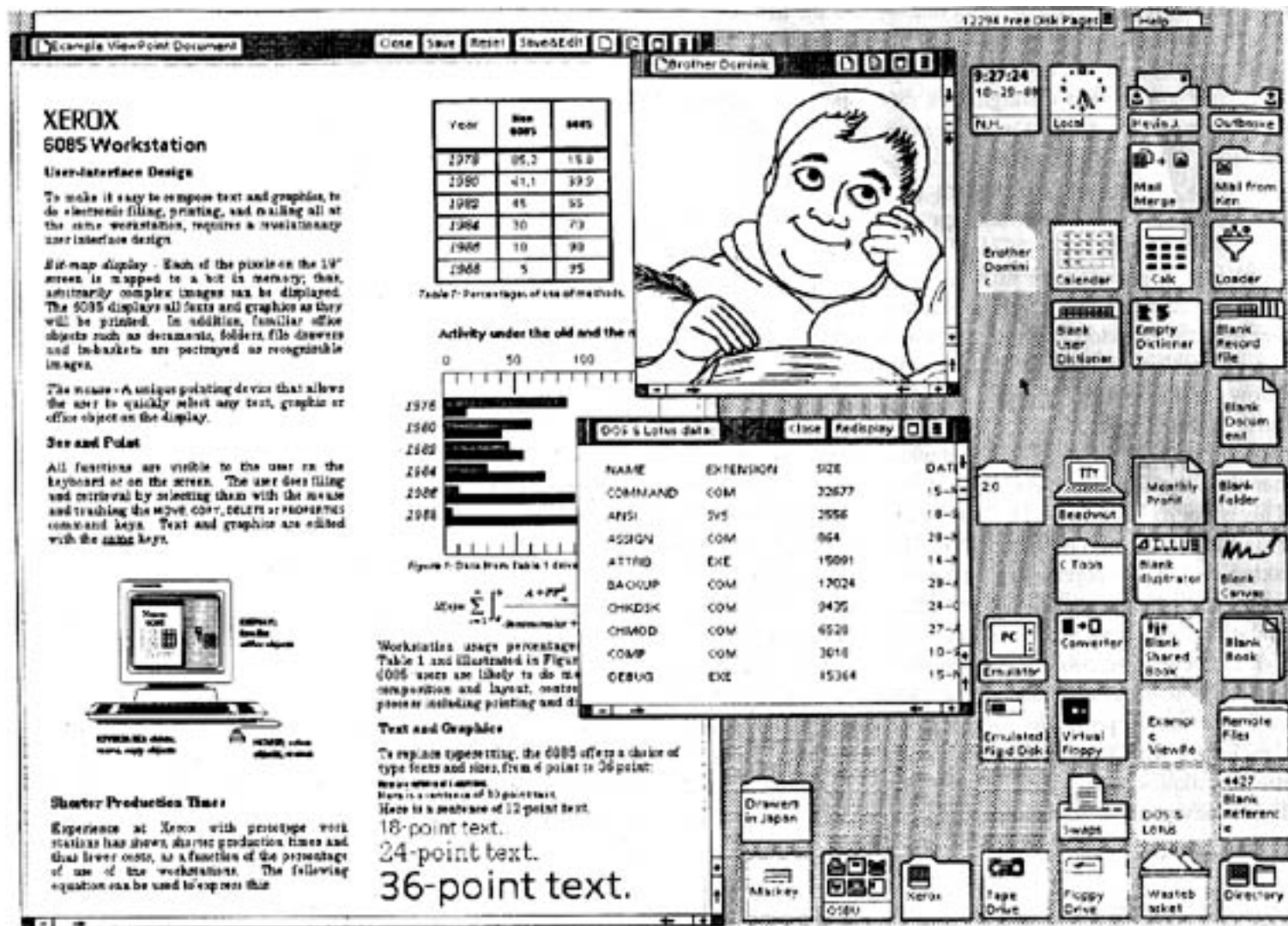
# Interface de comando do SO: CLI



OpenSUSE

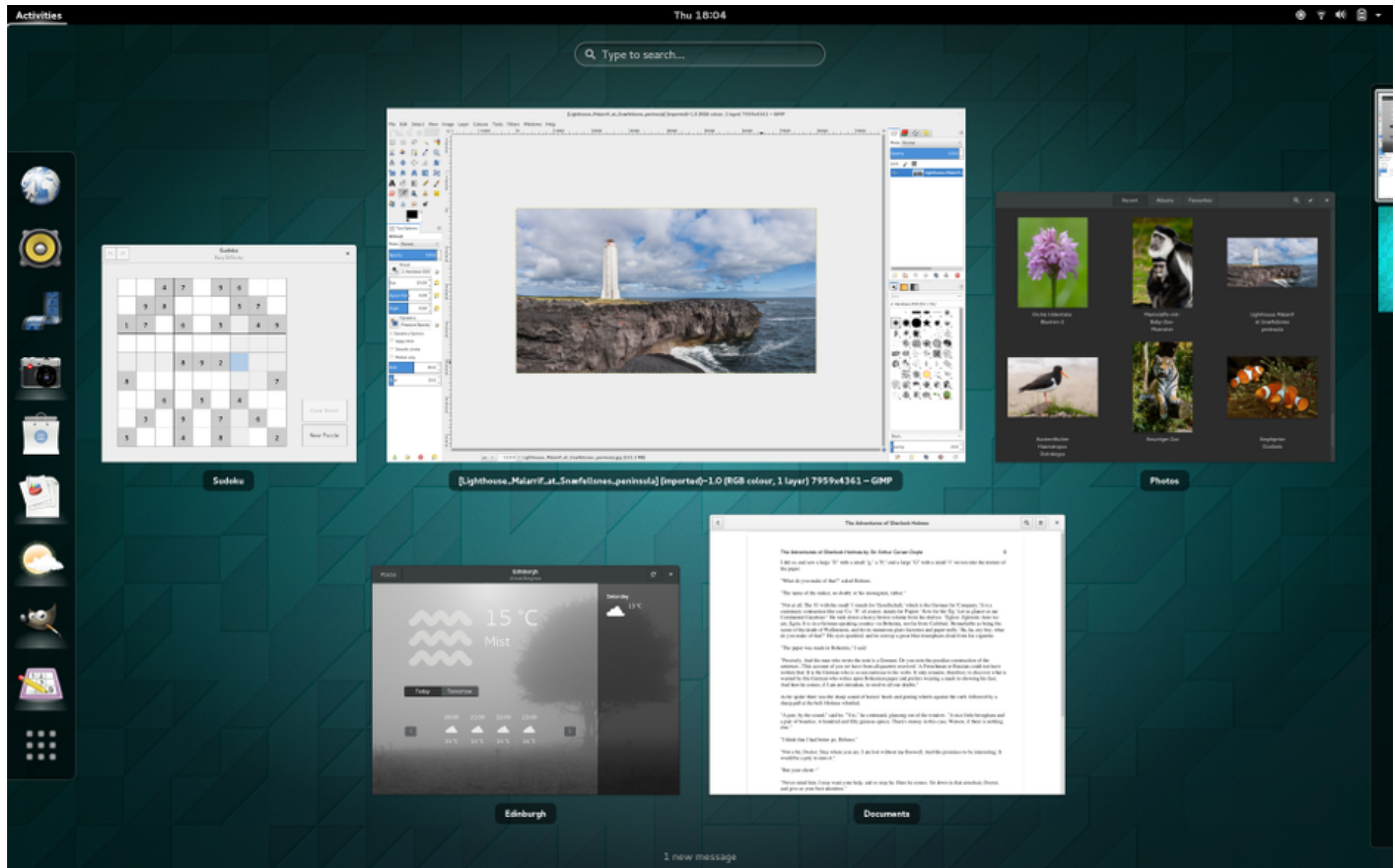
# Interface gráfica do SO: GUI

- **Interface amigável da metáfora do desktop**
  - Janela baseada em mouse e sistema de menus: Macintosh e Windows.
  - Ícones representam arquivos, programas e funções do sistema.
  - Inventado na Xerox PARC.



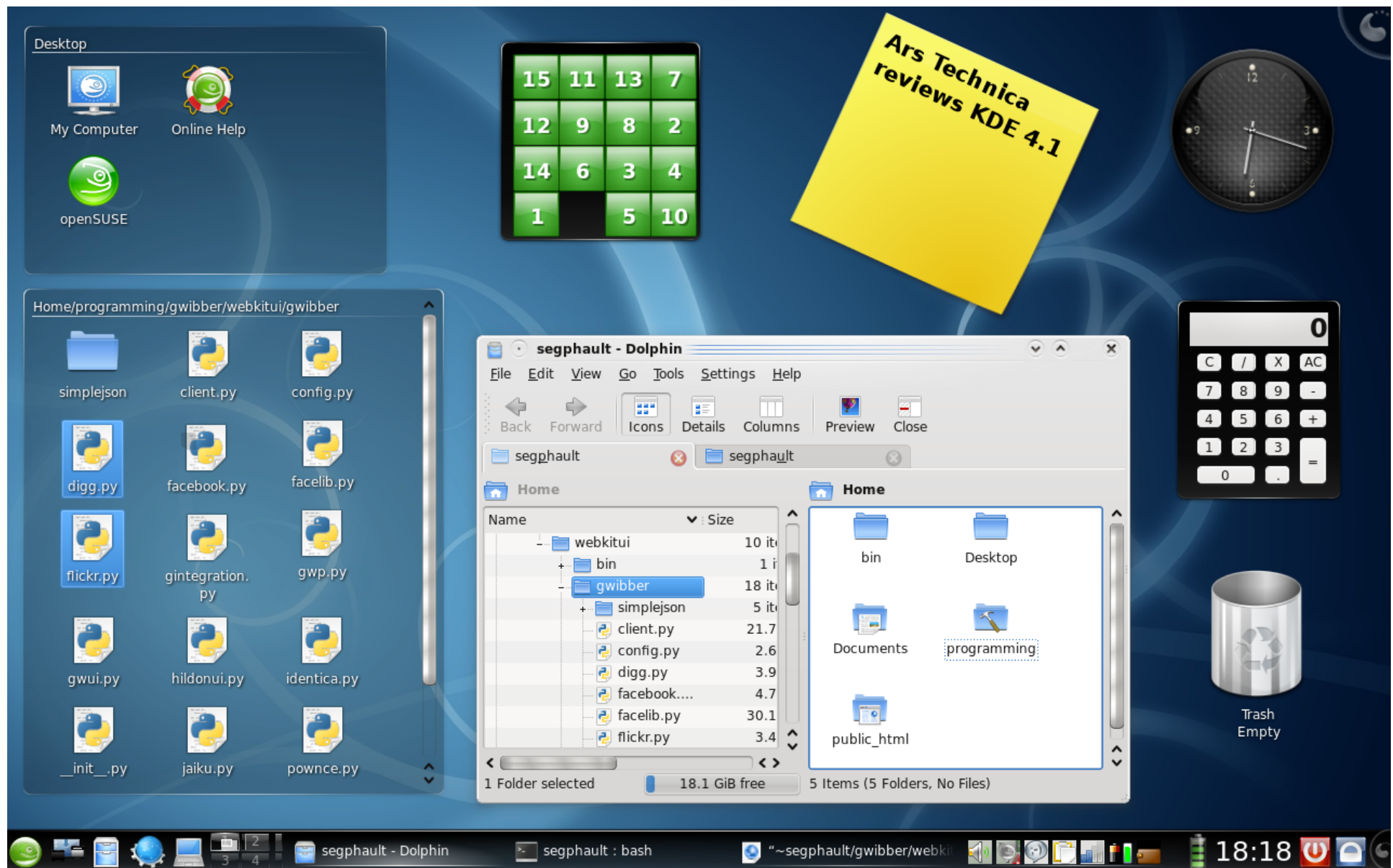
Xerox Star 1981





## GNOME 3

Sistemas Operacionais I – Prof. Diogo 2015-1



## KDE 4.1



# API (Application Programming Interface)

- **Definição**

- Conjunto de rotinas e padrões estabelecidos por um software que permitem que outro software utilize suas funcionalidades como serviços, sem se envolver em detalhes de implementação.
  - API: lista e descrição de funções oferecidas.
  - Biblioteca: conjunto de tais funções pré-compiladas.

- **APIs no SO**

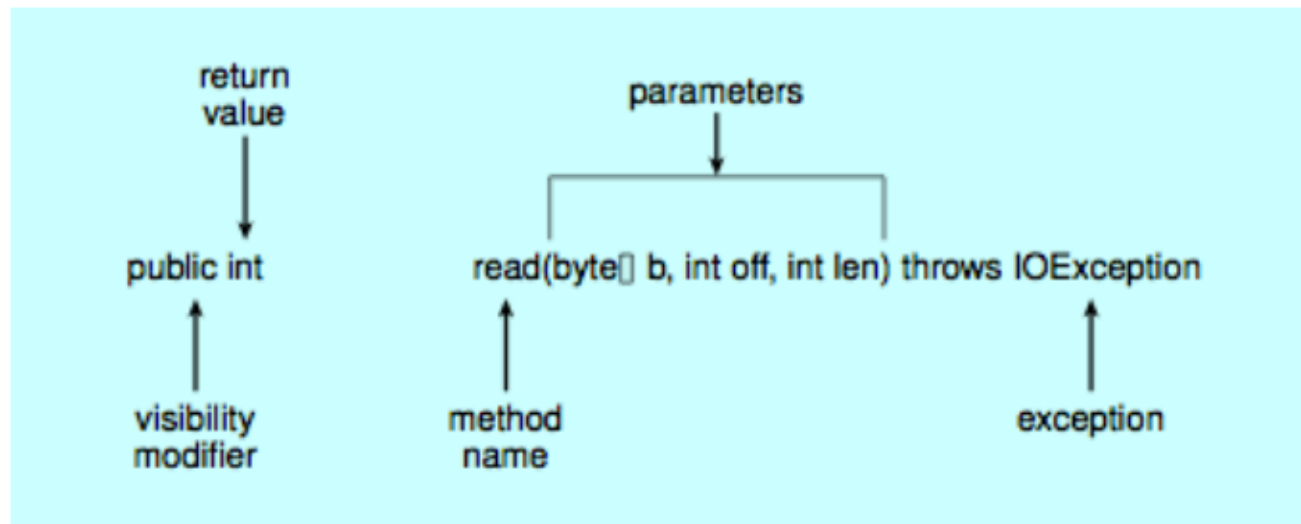
- Fornecem tarefas dissociadas das mais essenciais (manipulação de blocos de memória e acesso a dispositivos, que é feito pelo kernel).
  - Exemplos: criar janelas, acessar arquivos, criptografar dados...

# Chamadas de Sistema

- **Características**

- Proveem uma interface com os serviços de um SO.
- Normalmente escritas em C ou C++.
- Acessadas principalmente pelos programas por meio de uma API de alto nível, ao invés do uso da chamada direta de sistema.
  - Três APIs mais comuns são Win32 API para Windows, POSIX API para sistemas baseados em POSIX (incluindo praticamente todas as versões do UNIX, Linux e Mac OS X) e Java API para a Java Virtual Machine (JVM) → essa última é menos dependente do SO.

# Exemplo de Chamada de Sistema

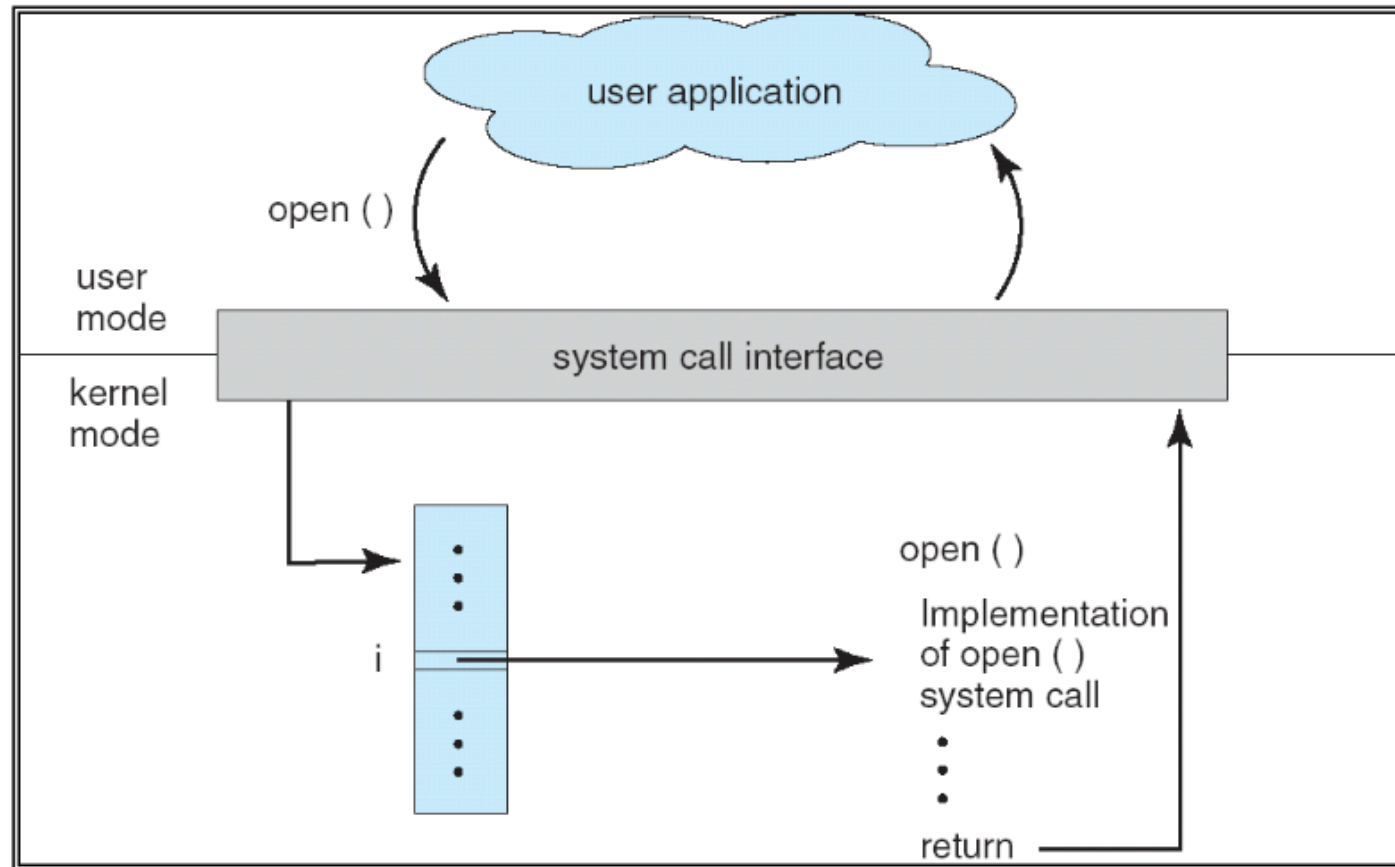


# Chamadas de Sistema

- **Identificação da chamada de sistema**
  - Cada chamada de sistema possui um número associado.
  - A interface de chamada do sistema mantém uma tabela indexada de acordo com esses números.
- **Funcionamento**
  - A interface de chamada de sistema invoca a chamada intencionada no kernel do SO e retorna o status da chamada com os valores de retorno.

# Chamada de Sistema

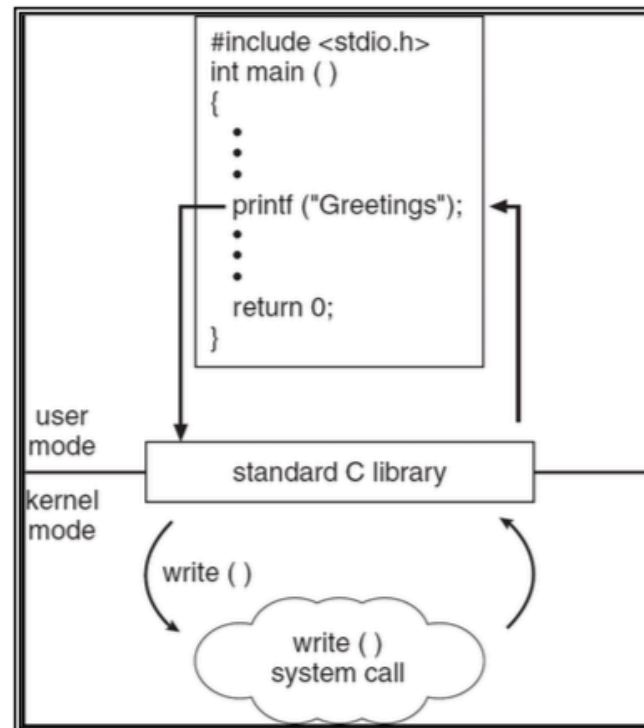
Fornece abstração e encapsulamento





# Exemplo de biblioteca C padrão

Programa C invocando chamada de biblioteca printf(),  
que faz a chamada do sistema write()



# Parâmetros na Chamada de Sistema

- **Chamadas e parâmetros**
  - Tipo exato e quantidade de informação variam de acordo com o SO e a chamada.
- **Métodos gerais usados para passar parâmetros ao SO**
  - Mais simples: via registradores.
  - Pilha: se registradores são insuficientes, usa-se a pilha.
  - Bloco: parâmetros armazenados em um bloco na memória.
    - O endereço do bloco é passado como um parâmetro em um registrador.

# Tipos de Chamadas de Sistema

- **Controle de processos**
- **Gerenciamento de arquivos**
- **Gerenciamento de dispositivos**
- **Manutenção de informações**
- **Comunicações**

# Sistema Monotarefa

- **Execução simultânea limitada**
- **Interpretador chamado na inicialização**
- **Usa o método simples de estender o SO: sem criar processo**
  - Escreve o programa na memória sobre parte de si.
  - Define o ponteiro de instruções para a primeira instrução.
  - O programa é executado até um erro (trap) ou fim (s-calls).
  - O interpretador retoma a execução.
  - Ele recarrega o restante de si.
  - Retorna ao usuário ou ao próximo programa.
- **Ex.: MSDOS**

# Sistema Multitarefa

- O shell escolhido pelo usuário é executado na inicialização
- O interpretador continua executando enquanto outros programas executam
  - Inicia um novo processo: `fork()`
  - Binário carregado na memória para execução pelo `exec()`
- O shell pode esperar o programa ou continuar executando
- O processo conclui: `exit()`
- Ex.: FreeBSD