



CONSULTORIA ESPECIALIZADA: SISTEMAS DISTRIBUÍDOS

## PARTE 02 - CONCEITOS BÁSICOS DE SISTEMAS DISTRIBUÍDOS E PROGRAMAÇÃO PARALELA

PROF. CARLOS PAES.



PUC-SP

# PROGRAMA



Conjunto de instruções (algoritmo) que determinam uma computação a ser realizada pelo computador



Solução para um problema que pode ser resolvido por um algoritmo



Entidade passiva! → arquivo executável armazenado em memória secundária (dispositivo de armazenamento secundário)

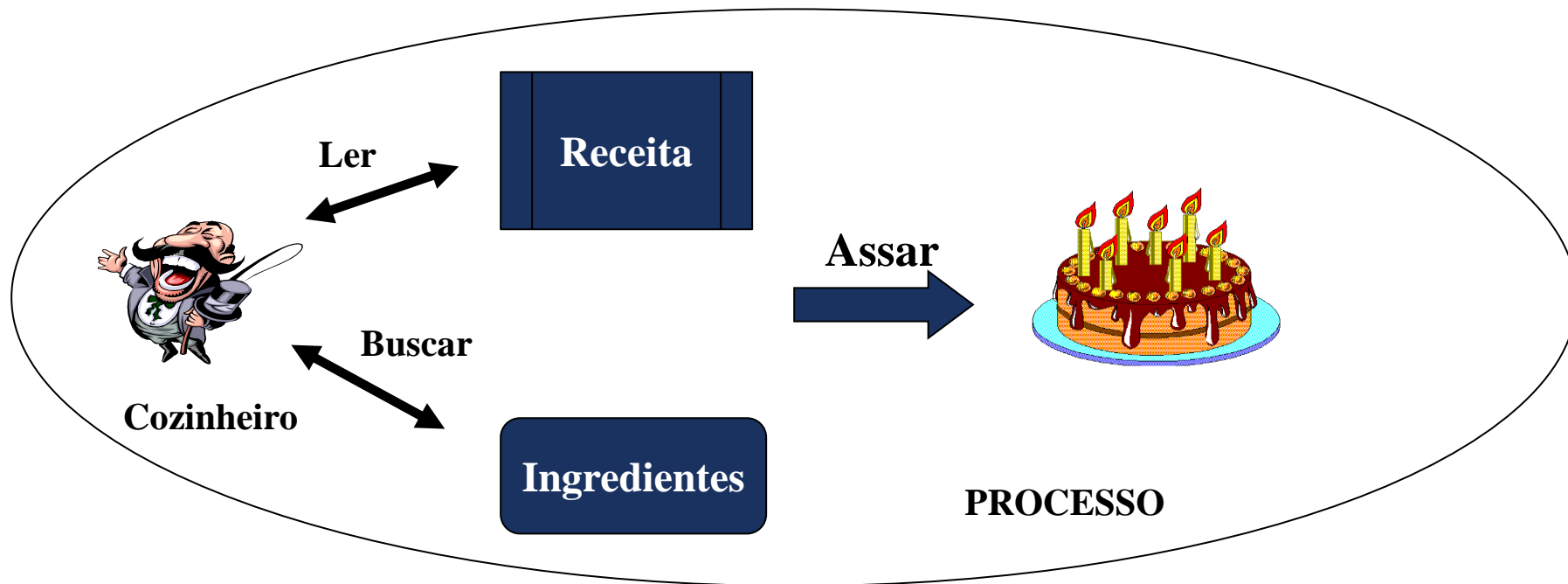


Vários tipos de programas: aplicativo, sistema, comercial, entretenimento e etc...

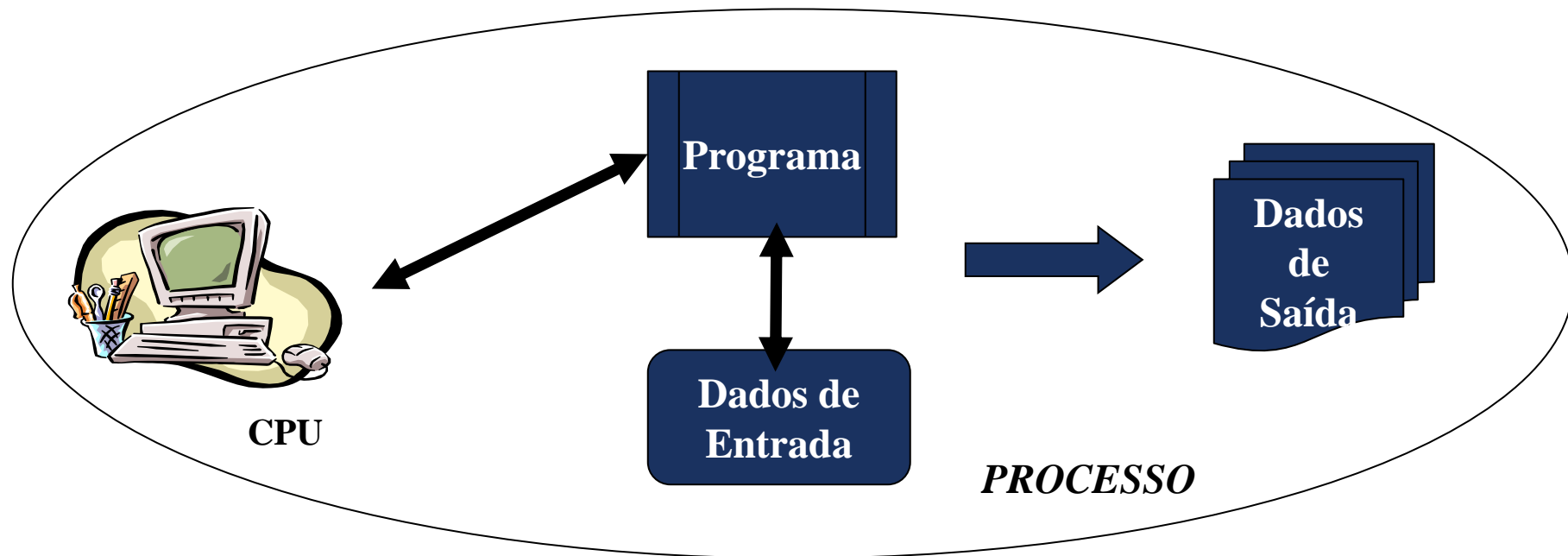
# PROCESSO

- Conceito fundamental em Sistemas Operacionais (multiprogramados/multitarefa)
- Um processo é um programa em execução, é uma entidade ativa
- Um programa não é um processo, é uma entidade passiva, uma sequência de instruções armazenadas em um arquivo em disco
- Embora 2 processos possam estar associados ao mesmo programa (reentrância), eles são 2 sequências de execuções concorrentes/paralelas
- Um processo pode criar outros processos enquanto executa

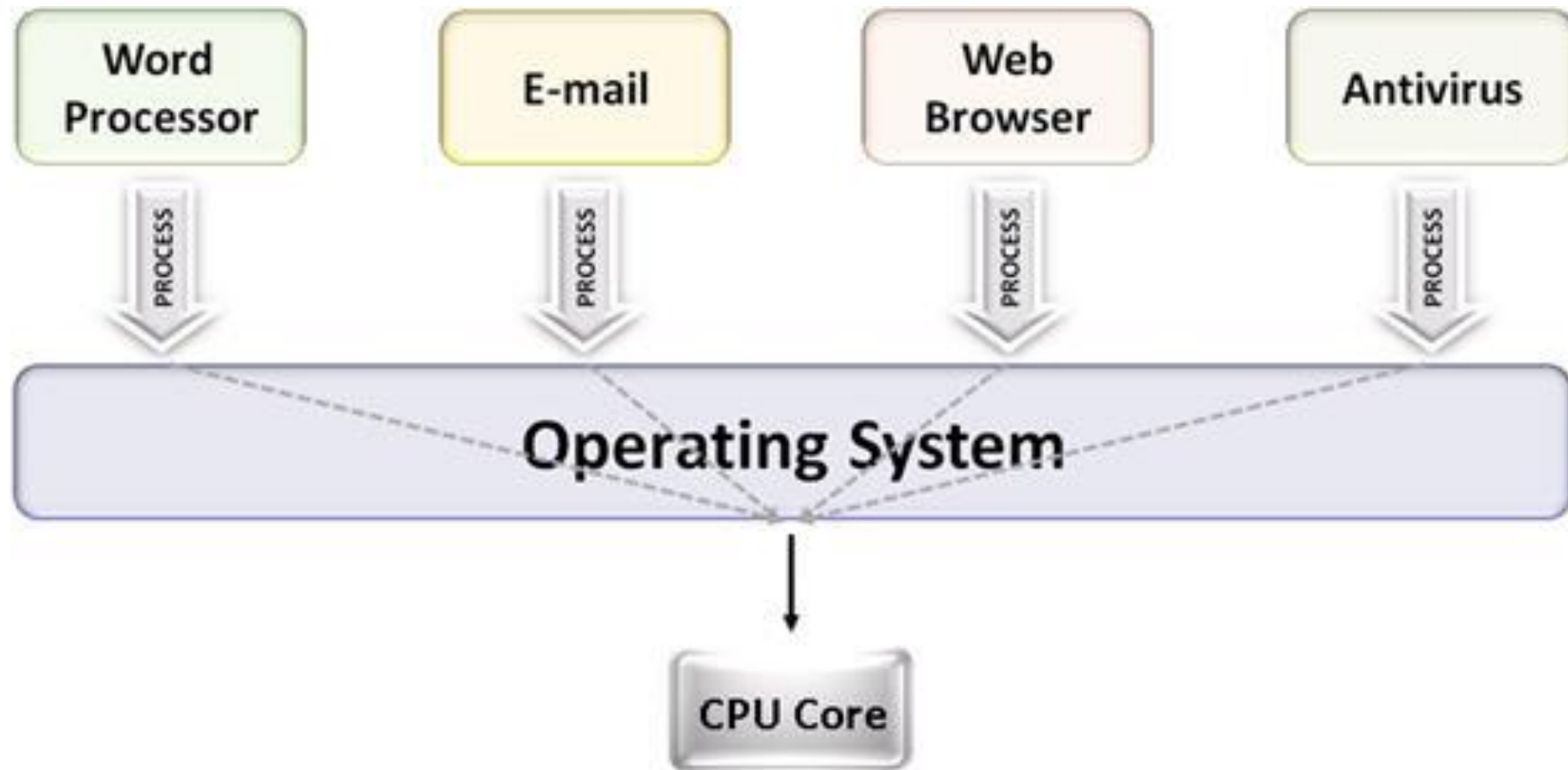
# PROCESSO



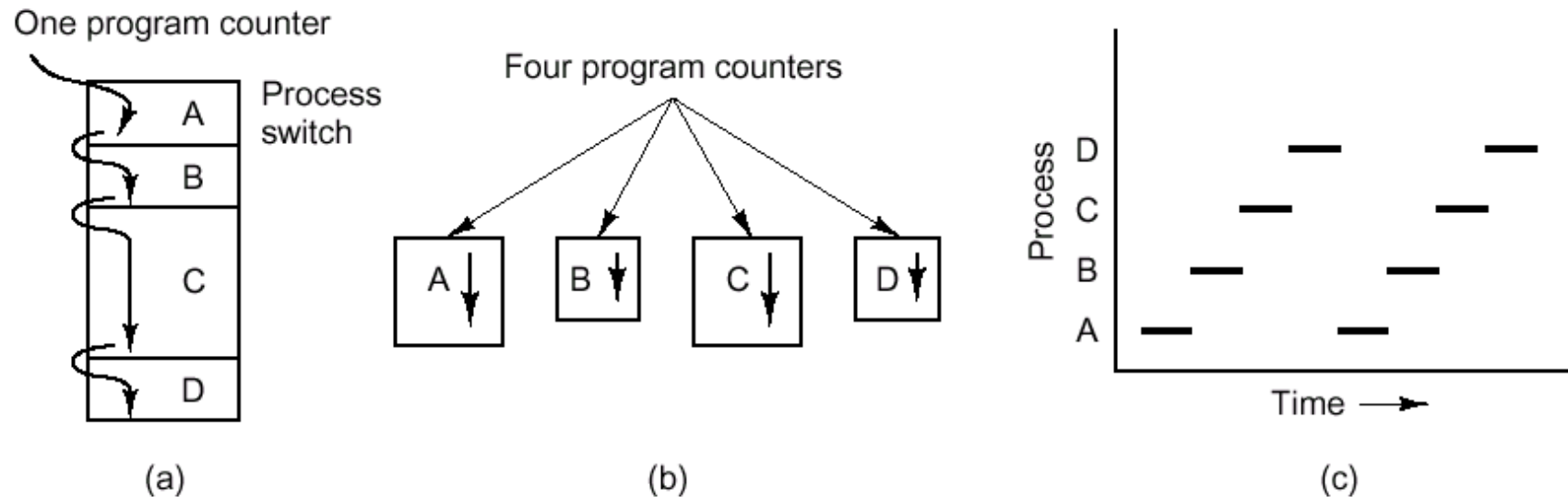
# PROCESSO



# MULTIPROGRAMAÇÃO

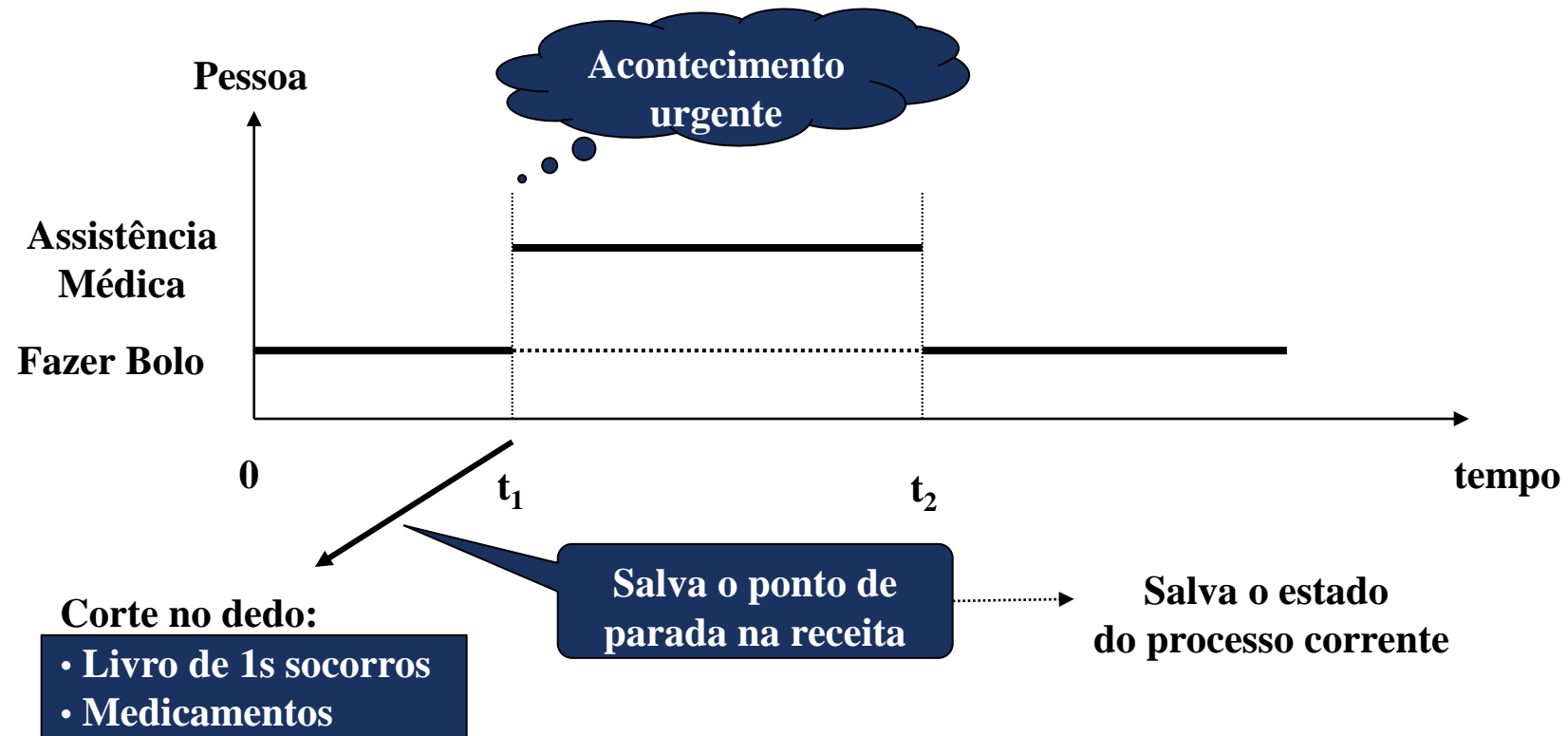


# MULTIPROGRAMAÇÃO



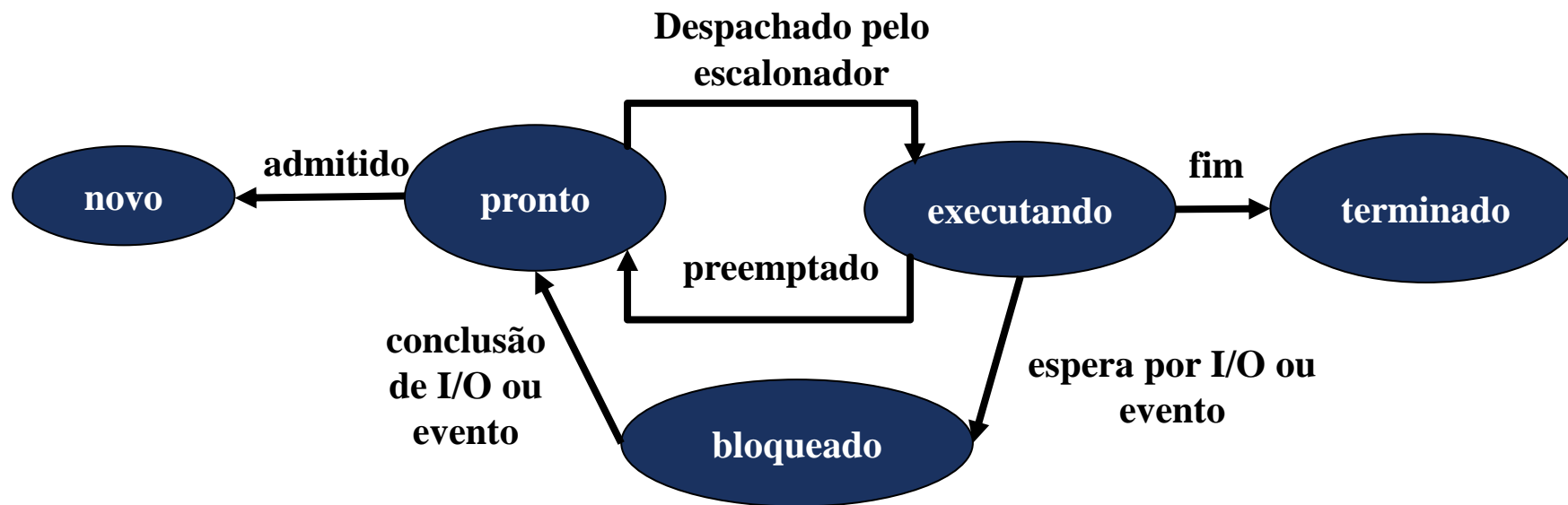
**Figure 2-1.** (a) Multiprogramming of four programs. (b) Conceptual model of four independent, sequential processes. (c) Only one program is active at any instant.

# MULTIPROGRAMAÇÃO





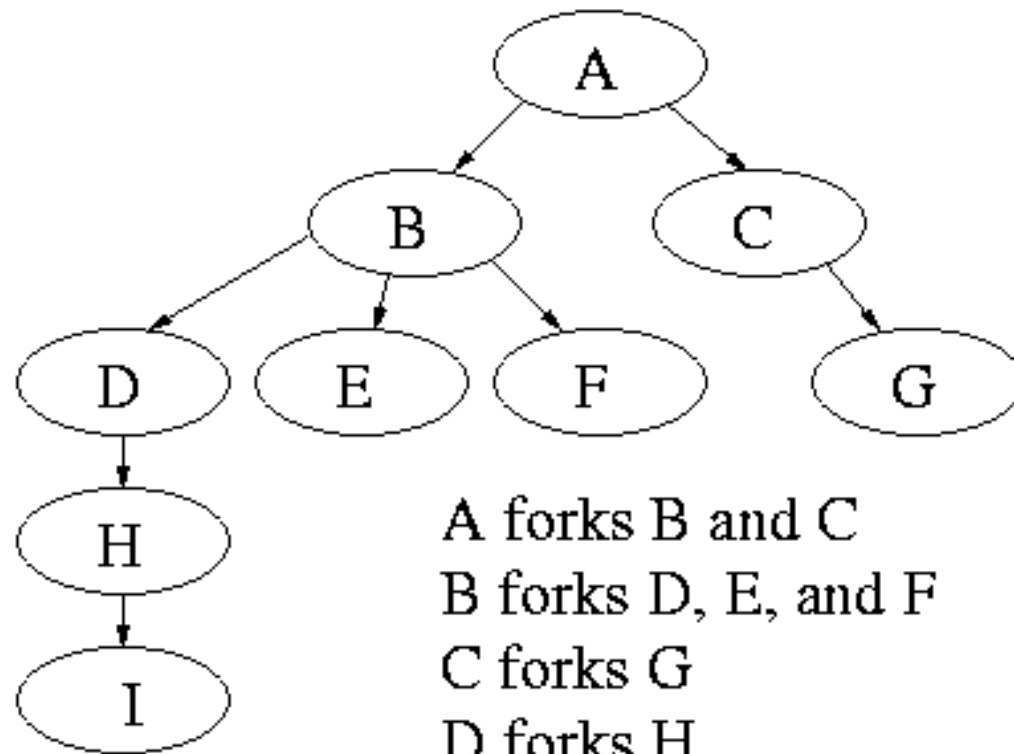
# ESTADOS DE UM PROCESSO NO SO



# CRIAÇÃO DE PROCESSOS

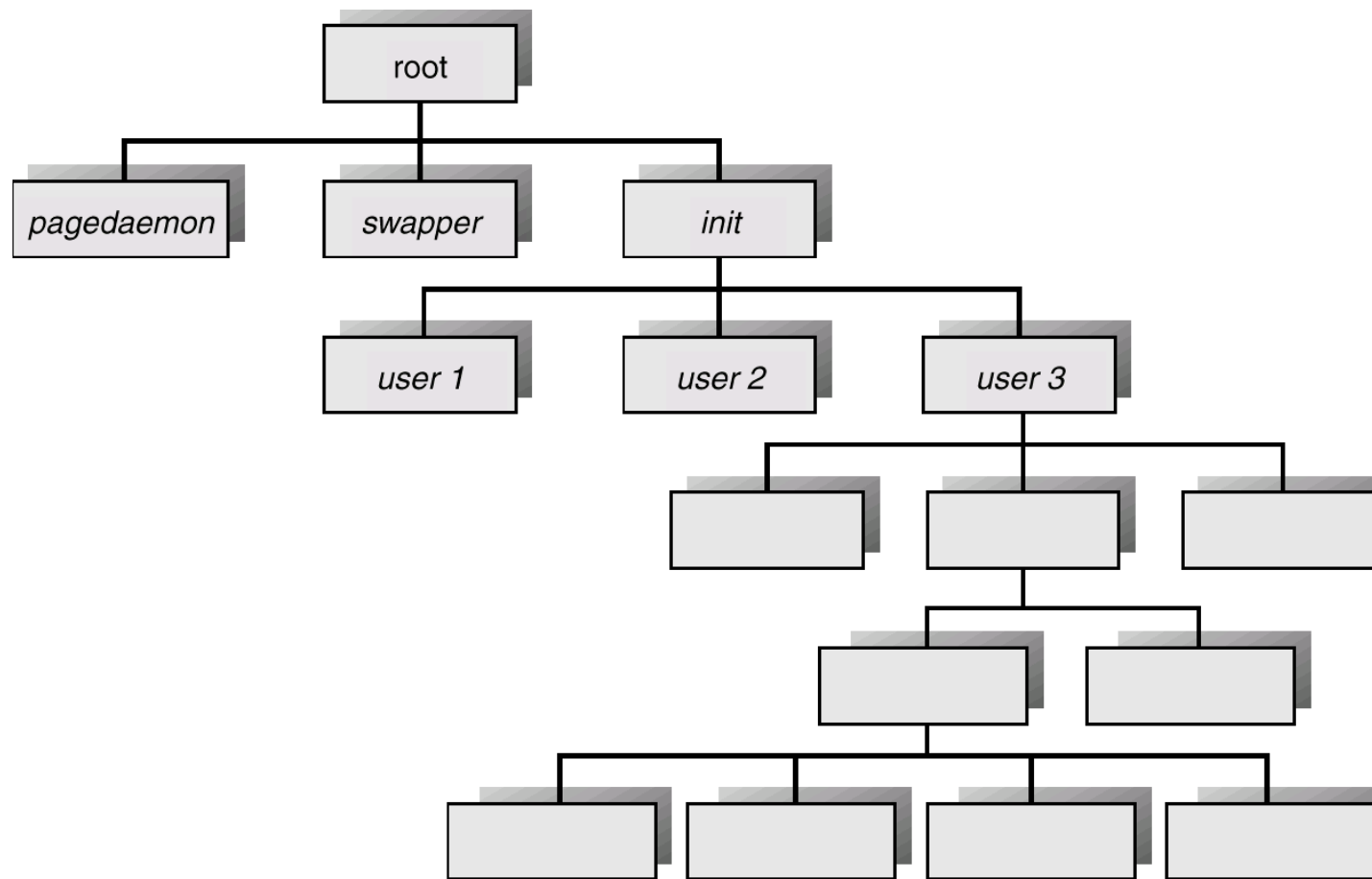
- Processo → durante sua execução pode criar novos processos por meio de chamadas ao sistema do tipo “create process”
- Processo que criou → chamado de processo pai
- Processo criado (novo processo) → chamado de processo filho
- Cada novo processo pode criar outros processos, formando uma árvore de processos

# CRIAÇÃO DE PROCESSOS



A forks B and C  
B forks D, E, and F  
C forks G  
D forks H  
H forks I

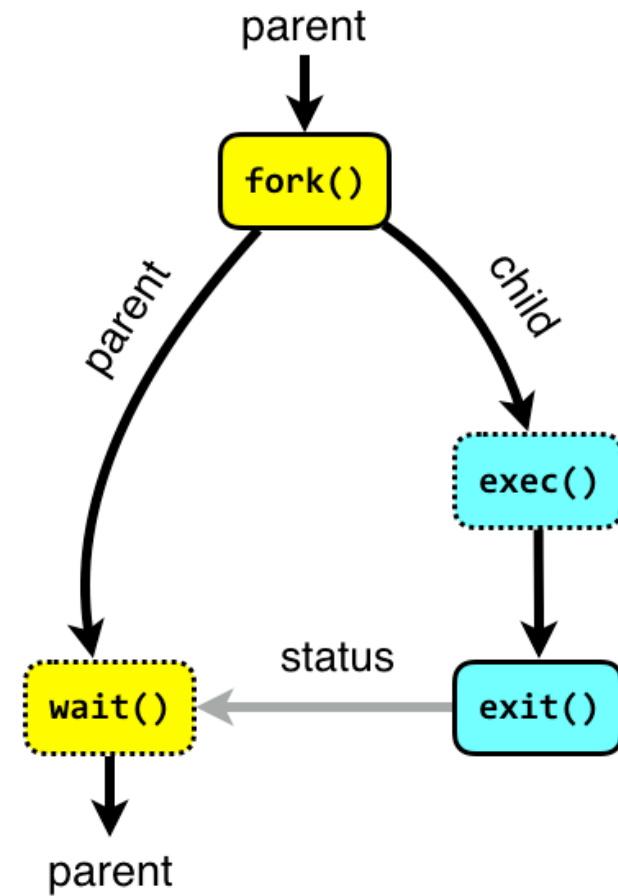
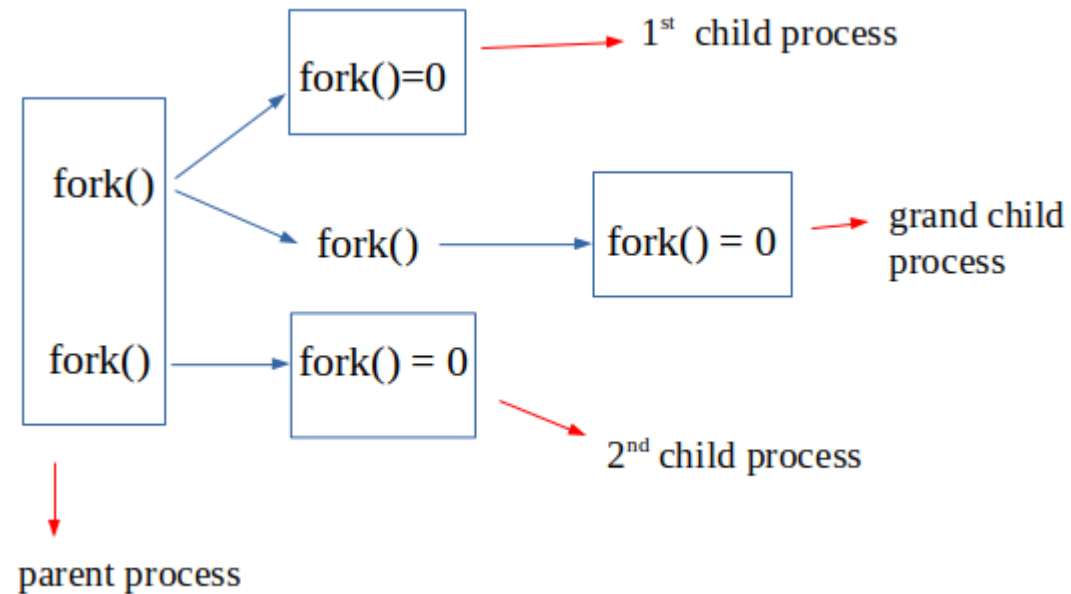
# ORGANIZAÇÃO DOS PROCESSOS NO SO (VISÃO UNIX)



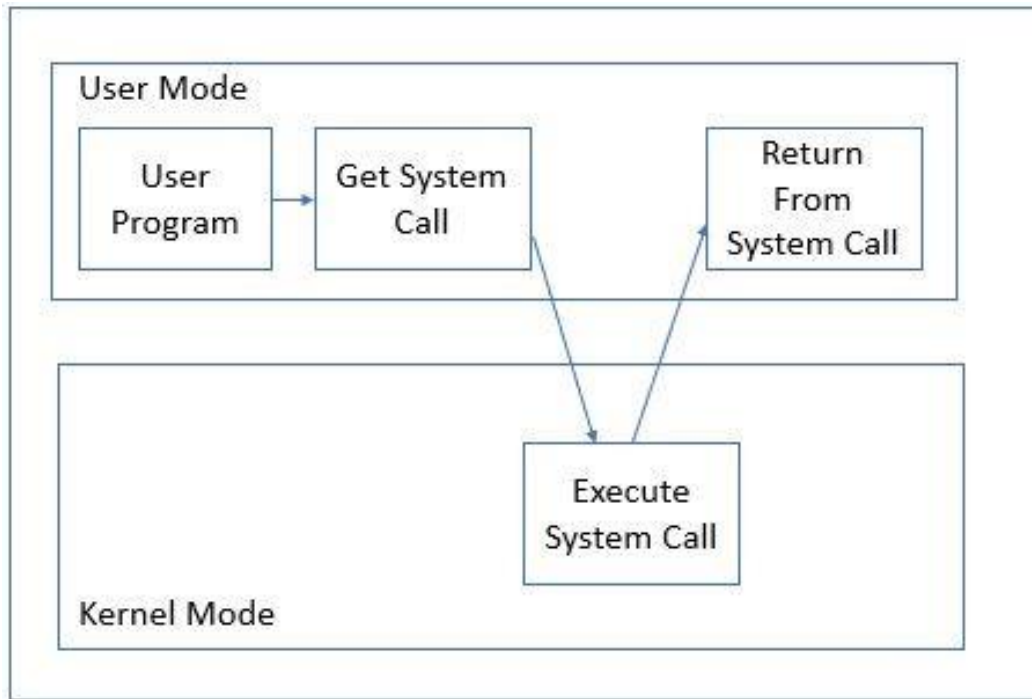
# CRIAÇÃO DE PROCESSOS

- Quando um processo  $P_p$  cria um novo processo  $P_f$ , em relação à execução de  $P_p$  e  $P_f$ :
  - $P_p$  continua a executar concorrentemente com seus processos filhos; ou
  - $P_p$  espera até que alguns ou todos os seus processos filhos tenham terminado
- Em relação ao espaços de endereçamento de :
  - $P_f$  é uma duplicação de ; ou
  - $P_f$  tem um programa diferente carregado nele

# CRIAÇÃO DE PROCESSOS



# CHAMADAS AO SO



- Interface de programação para serviços providos pelo SO
- São disponibilizadas como interrupção de software (baixo nível)
- Os SOs encapsulam as chamadas em bibliotecas na linguagem C/C++ (programação de sistema)

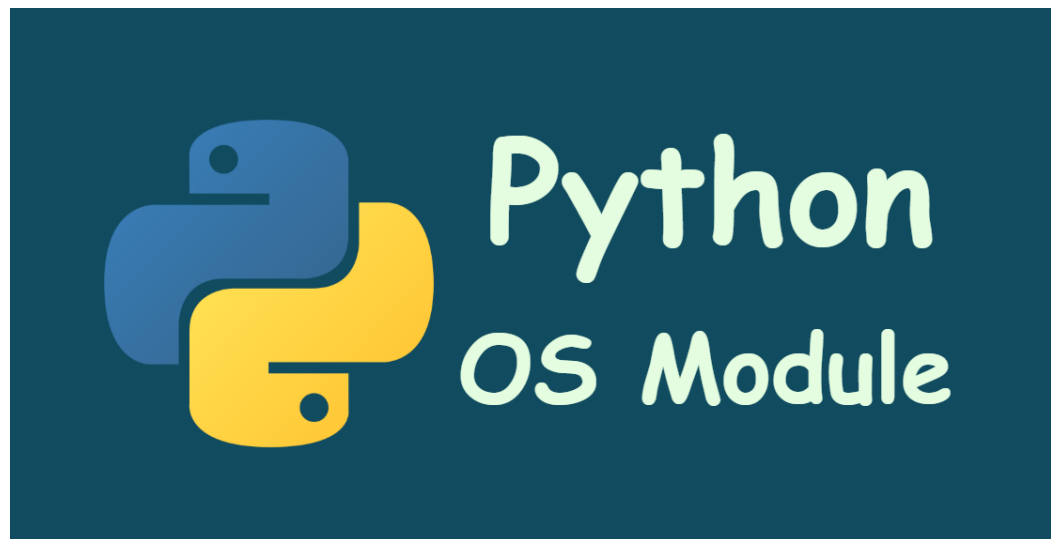
# CHAMADAS AO SISTEMA

- Mundo Unix (e Linux) → Systems Call
- Mundo Windows (Microsoft) → API (Application Program Interface)

(the users)		
shells and commands compilers and interpreters system libraries		
<i>system-call interface to the kernel</i>		
signals terminal handling character I/O system terminal drivers	file system swapping block I/O system disk and tape drivers	CPU scheduling page replacement demand paging virtual memory
<i>kernel interface to the hardware</i>		
terminal controllers terminals	device controllers disks and tapes	memory controllers physical memory

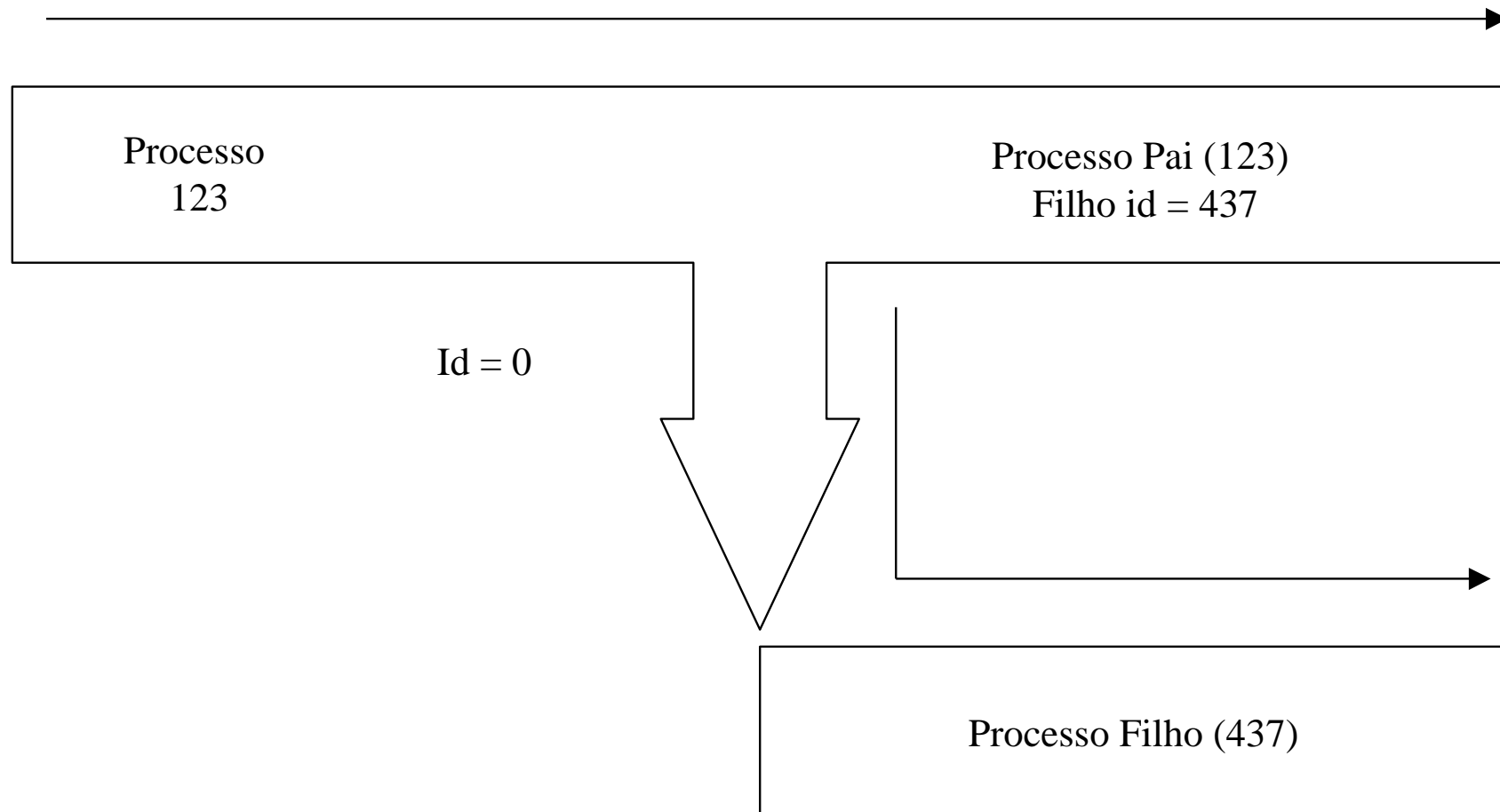


# COMO CHAMAR O SO EM PYTHON

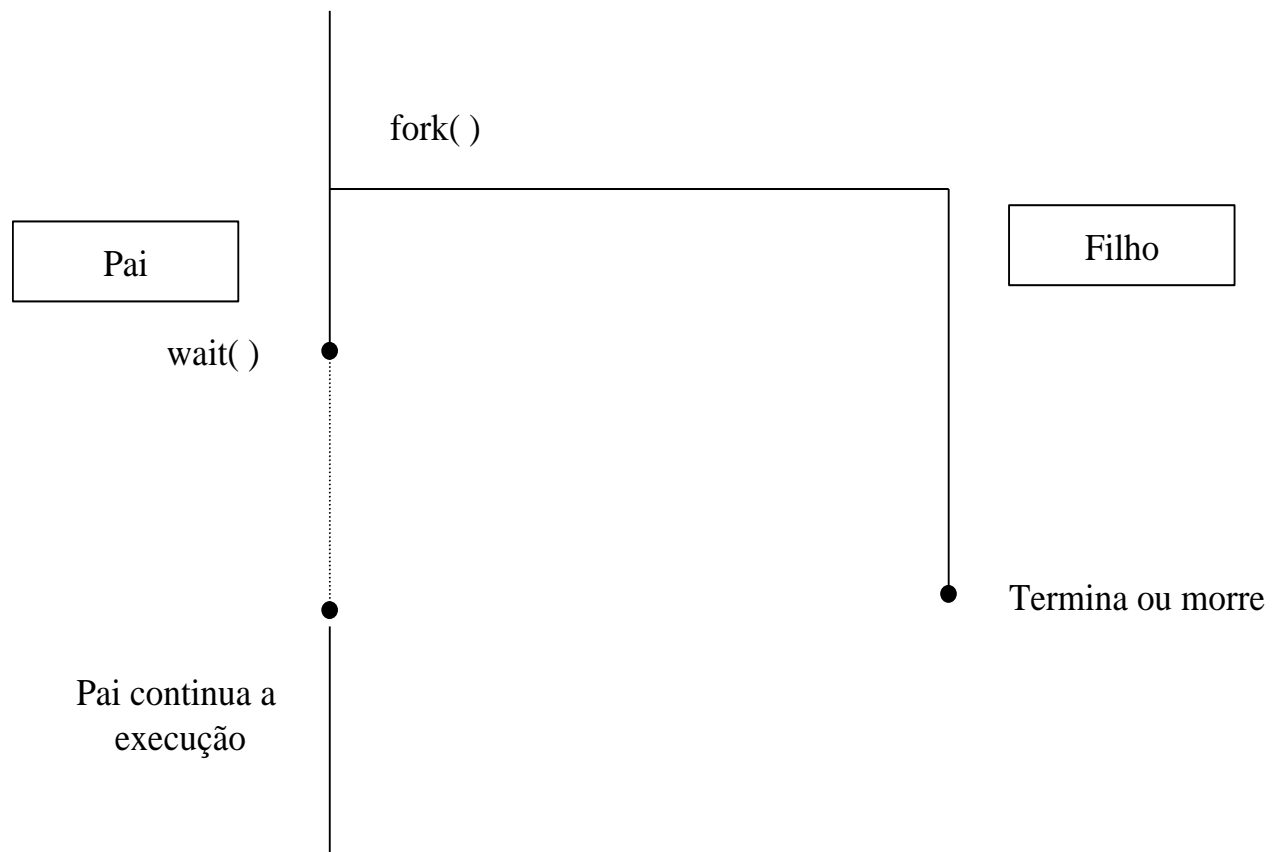


- Módulo OS do Python → Fornece uma maneira portátil de usar a funcionalidade dependente do sistema operacional
- Principais funcionalidades: gerenciamento de processos, sistema de arquivos e etc..
- <https://docs.python.org/3/library/os.html>

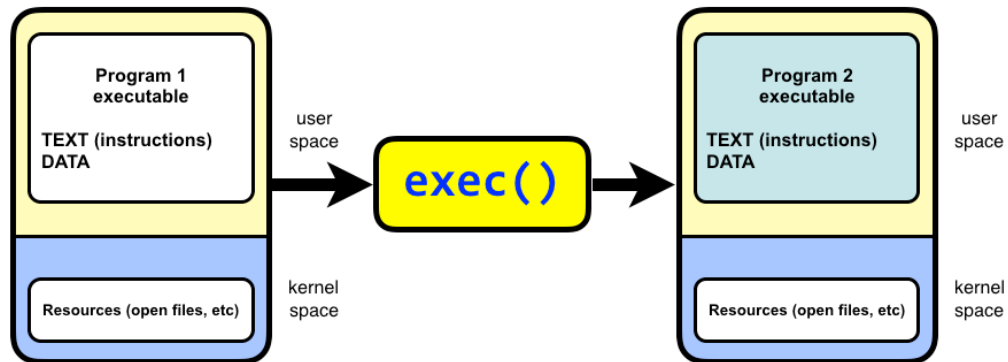
# CRIAÇÃO DE PROCESSO (FORK)



# CRIAÇÃO DE PROCESSO (FORK E WAIT)



# CHAMADA EXEC



- Faz outro programa executar
- Substitui o processo que chama `exec` por um novo programa e começa a executar o novo programa
- PID não muda
- `exec` substitui segmentos de texto, dados, heap e pilha por novo programa lido do disco; caso não encontre retorna -1



# OBRIGADO

CARLOSP@PUCSP.BR