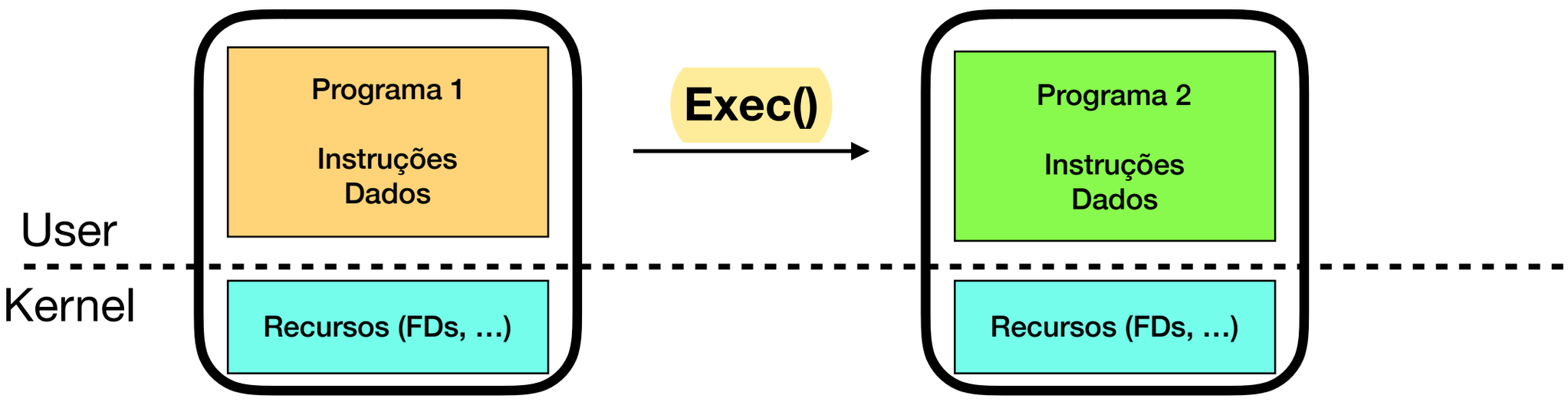
Sistemas Operativos

Exec

- A familia de funções exec substitui o programa (imagem do processo) em execução num certo processo pelo novo programa (nova imagem do processo) especificado como argumento

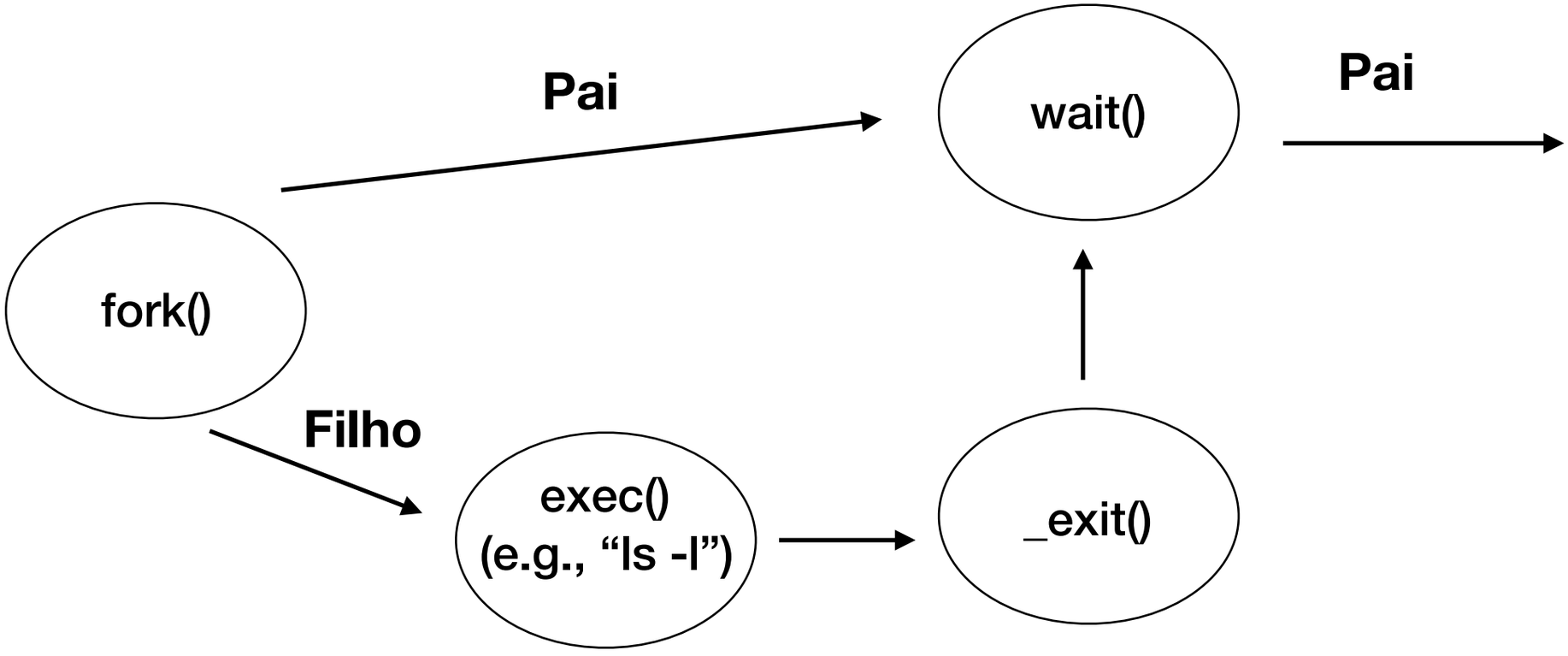
- Instruções e dados (p.ex: argumentos) são substituídos

- Pid, descritores e estruturas em kernel são mantidos



Utilidade

- Fork + Exec - **interpretador de comandos** - processo pai cria um filho via o comando “fork()” e o filho executa um novo programa via o comando “exec()”



Chamadas ao sistema

- Bibliotecas
  - <unistd.h> - definições e declarações de chamadas

Chamadas ao sistema

- int execl(const char *path, const char *arg0, ..., NULL);
    - troca o programa em execução pelo executável definido pelos argumentos da função
    - **path** - caminho completo para o executável
    - **arg0** - nome do executável (segue as normas unix em que o arg[0] é sempre o nome do executável)*
    - **arg1, ..., NULL** - argumentos do executável. O último argumento tem de ter o valor NULL
    - apenas retorna valor em caso de erro

*O nome do executável pode ser alterado (ver exercício 4 do guião 3)

Chamadas ao sistema

- int execlp(const char *file, const char *arg0, ..., NULL);
  - Semelhante à chamada anterior mas recorre aos caminhos registados na variável de ambiente *PATH*
  - Exemplo: em vez de “/bin/ls” o argumento **file** poderia ser apenas “ls”

Chamadas ao sistema

- int execv(const char *path, char *const argv[]);
- int execvp(const char *file, char *const argv[]);
  - Semelhantes às chamadas anteriores mas recebe argumentos do executável como um array (**argv**)
  - Primeira posição do array **argv** deve conter o nome do executável. A última deve conter o valor NULL

Material de Apoio

- http://www.it.uu.se/education/course/homepage/os/vt18/module-2/exec/