

Sistemas Operativos

Primitivas em Unix

Primitiva *fork* e outras



- *Fork* é invocado uma vez pelo pai mas retorna duas vezes:
 - uma no pai e outra no filho;
 - no pai, *fork* retorna o PID do filho criado;
 - no filho, *fork* retorna zero;
 - se mal sucedido, *fork* retorna -1 no pai
- Em ambos os processos, a execução do programa prossegue na instrução imediatamente a seguir ao *fork*
- Outras primitivas:
 - *getpid* e *getppid* são primitivas para obter o PID e o PPID, respectivamente, do processo invocador
 - Um processo deve invocar a primitiva *exit* para terminar, na qual deve ser indicado o estado de saída do processo (0 se o processo termina normalmente)
 - O processo pai tem acesso ao estado de saída do filho através da primitiva *wait*. Esta primitiva bloqueia um processo até que um filho qualquer termine.

Exercício



- Escreva um programa em C que crie dois processos. O filho deve mostrar o seu PID e o seu PPID. O pai deve apresentar o seu PID.

3

Luísa Jorge

Sistemas Operativos

Resolução



```
#include <stdio.h>
#include <sys/types>
#include <unistd.h>
main(){
    pid_t pid;
    pid=fork();
    if(pid<0){
        perror("fork");
        exit(1);
    } else if (pid==0){
        printf("Sou o filho %ld do pai %ld\n",getpid(), getppid());
        exit(0);
    }
    printf ("sou o pai %ld\n", getpid());
    exit(0);
}
```

4

Luísa Jorge

Sistemas Operativos