西安交通大学
软件学院

操作系统原理

Operating System Principle

田丽华

3-6 进程操作

UNIX examples UNIX例子

- 在UNIX系统中用户键入一个命令（如date, ps,ls），shell就创建一个进程。
- **fork** system call creates new process

  fork 系统调用创建新进程
- **execlp** system call used after a **fork** to replace the process' memory space with a new program.

  在**fork**之后采用**execlp**系统调用用一个新程序替代进程的内存空间

Process Creation (Cont.)

进程创建

```
#include<stdio.h>
Void main(int argc, char *argv[])
{ int pid;
   pid = fork(); /*fork another process*/
   if (pid < 0) {/* error occurred */
      fprintf(stderr, "Fork Failed");
      exit(-1); }
   else if (pid = 0) { /* child process */
      execlp("/bin/ls","ls",NULL);
    } else {   /* parent process */
      wait(NULL);
      printf("Child Complete");
      exit(0);
    }
}
```

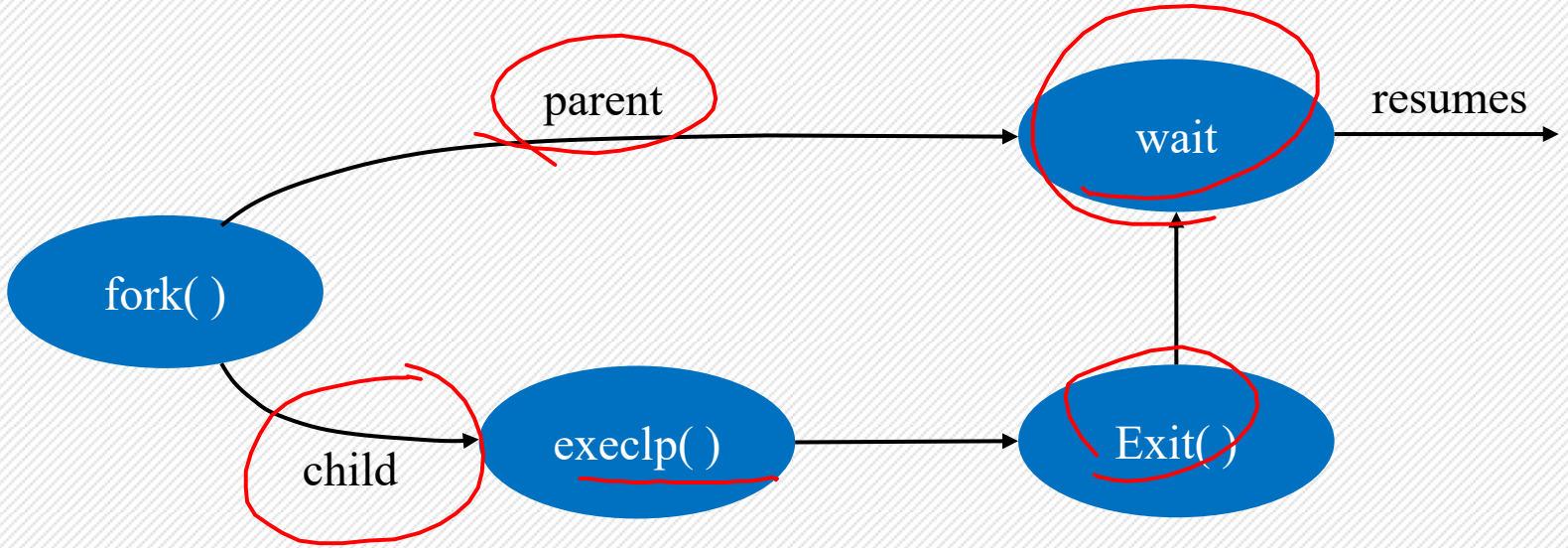
UNIX系统

pid=fork();

fork之后，操作系统会复制一个与父进程完全相同的子进程。

从系统调用fork返回时，CPU在父进程中时，pid值为所创建子进程的进程号（>0），若在子进程中时，pid的值为零。

Process Creation



Process Termination
进程终止

- Process executes last statement and asks the operating system to decide it (exit).进程执行最后一项并询问操作系统作出决定（退出）

  Output data from child to parent (via wait).

  从子进程向父进程输出数据）（通过等待）

  - Process' resources are deallocated by operating system. 操作系统收回进程的资源
- Parent may terminate execution of children processes (abort).

  父进程可中止子进程的执行（终止）

  - Child has exceeded allocated resources.子进程超量分配资源
  - Task assigned to child is no longer required.

    赋予子进程的任务不再需要
  - Parent is exiting. 父进程终止
    - Operating system does not allow child to continue if its parent terminates. 若父进程终止，不允许子进程继续
    - Cascading termination. 级联终止

Process Blocking
进程阻塞

一个处在运行状态的进程，因等待某个事件的发生（如等待打印机、同步事件等）而不能继续运行时，将调用阻塞原语，把进程置为阻塞状态，并转进程调度程序（等于让出处理机）。

调用进程阻塞操作是在进程处于运行状态下执行的。它的执行将引起等待某事件的队列的改变。

进程唤醒

当进程所等待的事件发生时，该进程将被唤醒(由进程唤醒操作完成)。

1.

2.

唤醒一个进程有两种方法：

- 由系统进程唤醒。
- 由事件发生进程唤醒。