

OS HW02

1. $1 + 1 + 2 + 4 + 8 = 16$;

- keep splitting like a binary tree;

2. fork successes and execlp fails;

- fork must succeed;
- if exec() successes, it will start a new program and will never return;

3. A is 0, B is 2603, C is 2603, D is 2600;

- A is the return value stored in pid of fork(), which is zero;
- B is the pid of current child process;
- C is the actual pid of child process;
- D is the pid of parent process;

4. x : 0, -1, -4, -9, -16 and y : 0, 1, 2, 3, 4

- child process copies the data of parent process but won't affect parent process. It's just a copy.

5. two cases

- if exec() successes, it won't return so that LINE X won't be executed;
- else if exec() fails, LINE X will be executed;

6.

- terminated status(ts) 可以被子进程用于向父进程反馈信息：
 - 运行结果;
 - 结束方式;
 - 发出信号SIGCHLD以唤醒被挂起的父进程。
- 藉由此，父进程可以做出相应的相应，并且可以控制父子间的执行顺序（通过wait()和ts所发出的相关信号）。

7.

- 僵尸进程是指当进程已经终止达但是其父进程尚未调用wait()的进程;
- 这种进程状态一般而言只是短暂存在，一旦父进程调用wait(), 其进程的进程标识符和它在进程表中的条目就会释放并最终被消灭掉。

8.

- 用户空间：用户程序的代码和数据（局部/全局变量，常量等）；
- 内核空间：系统调用的代码和数据，PCB（程序ID，进程状态等）；

9.

- exec()会破坏原有的包含系统调用exec()的原来程序的内容，调用新程序覆盖进程地址空间，且不会返回（除非失败）；
- 普通的调用则不会改变正在执行的程序。

10.

- 优点：
 - **响应性**：即使部分阻塞或执行冗长操作，仍可以继续执行；
 - **资源共享**：允许一个应用程序在同一地址空间内有多个不同活动线程；
 - **经济**：线程能够共享所属进程的资源，因此创建和切换线程更加经济；
 - **可伸缩性**：线程可以在多处理核上并行运行。
- bc

11.

- 5. First fork() creates process 1; In second fork() process 1 creates process 2; In third fork() original process and process 1 and process 2 create 3 more processes.
- 2. Process 1 and 2 create threads.

12. c is 5, p is 0

- 同一进程下的线程共享全局变量value，故c为5；
- 而父进程与子进程并不共享，只是复制，故为0；

13.

- 普通管道是单向的而命名管道可以双向；
- 普通管道需要进程有父子关系而命名管道无此限制；
- 普通管道在进程完成通信且终止后就不再存在而命名管道继续存在；
- 普通管道只可用于两个进程而命名管道可用于多个进程。