E11：UNIX V6++的进程创建与父子进程同步

参考答案与说明

1. 【参考答案】

父进程创建子进程时,如果内存空间不足,子进程的图像将被创建在盘交换区上。此时，子进程的状态为： p_stat=SRUN，p_flag=~SLOAD|SSWAP。

子进程作为一个在盘交换区上就绪的进程，未来会在某一个时刻被 0#进程将图像调入内存。子进程未来如果在 Swtch 中被 0#进程选中，将依次完成如下工作（详见 Swtch 函数流程）：

（1）在 swtch 中，根据相对虚实地址映射表构建两张用户页表；

（2）因为 SSWAP 的标志位被设置，用子进程的 u_ssav 数组再次恢复现场；

（3）执行 swtch 中的 return 1 指令，从核心栈栈顶弹出返回地址（NewProc 的下一条指令地址），带回返回值 1；

（4）转去执行 NewProc 的下一条指令。

所以，子进程从 Swtch 返回后，将执行 NewProc 的下一条指令。

2. 【参考答案】

fork()之后，父进程执行语句为：

```
    {
            a=a+1;
            printf(" i= %d; a= %d\n", i, a);
    }
    printf("The Tail.\n");
}
```

子进程执行语句为：

```
    {
            a=a+2;
            printf(" i= %d; a= %d\n", i, a);
    }
    printf("The Tail.\n");
}
```

程序的输出结果可能有下列四种情况：

（1） i=505； a=1
The Tail.
i=0; a=2
The Tail.

（2） i=0； a=2
The Tail.
i=505; a=1

The Tail.

(3) i=505；a=1
i=0; a=2
The Tail.
The Tail.

(4) i=0；a=2
i=505; a=1
The Tail.
The Tail.

3. 【参考答案】程序的输出结果如下：
It is child process.
It is parent process.
The finished child process is 505.
The exit status is 1.

终止码的传送过程如下：

① 子进程将在执行系统调用 exit 的过程中，借助现场保护，将终止码 1，压入其核心栈中保护 EBX 单元；
② 通过系统调用的参数传递，终止码 1 由核心栈中保护 EBX 单元送入子进程的 user 结构中的 u_arg[0]；
③ 在执行内核函数 Process::Exit 的过程中，user 结构被暂存在盘交换区上；
④ 父进程执行 wait 系统调用的过程中，借助现场保护，将变量 j 的地址，压入其核心栈中保护 EBX 单元；
⑤ 通过系统调用的参数传递，变量 j 的地址由核心栈中保护 EBX 单元送入父进程的 user 结构中的 u_arg[0]，则父进程 u_arg[0]指向变量 j；
⑥ 在执行内核函数 Process::Wait 的过程中，从磁盘将子进程的 user 结构读入一个内存缓存，并将其中子进程 u_arg[0]单元中保存的终止码 1 写入父进程 u_arg[0]指向的内存单元即变量 j。