一、名词解释

1、符号链接

2、控制终端

3、守护进程

4、全缓冲IO

二、问答题

1、如何写一个守护进程？

2、有人说，管道（pipe）不仅是两个进程间用来传递信息的方法，而且是两个进程间同步或互斥的一种方法。你认为这种说法对不对？为什么？

3、请用一条C语句实现这个功能：有一个字符型的数组，往其中写入这么一个字符串：“file.n.m”，其中n是进程的进程号，m是进程的真实用户ID号。

比如一个进程的进程号是1111，真实用户ID是2222，那么字符数组中应为“file.1111.2222”。注意，不必写任何需要包含的头文件。

4、请写出在网络通信时，服务器端进程对下列函数的执行顺序：

（1）connect；(2)close；（3）reads/writes；（4）socket；（5）bind；（6）accept；（7）listen。

Socket()---bind()---listen()---accept()---reads/writes----close()

三、辨析题

请说明下面的说法对或错，并给出理由。

1、在同一进程的不同线程中，一个线程可以访问其他线程的栈内容；而在不同进程的线程中，则不可互相访问各自的栈内容。（考完问老师，应该是对的）

2、如果一个线程在临界区发生了异常，那么OS会终止掉该线程，并且会释放掉该线程占有的锁，这样，其他等待该锁的线程就可以继续执行临界区代码而不会产生错误。

（假设OS知道所有的锁都被哪些线程占用了）。

3、一个线程调用了pthread\_exit后，操作系统在切换到其他线程前，它的栈空间应该被释放。

四、读程序，写出最终结果，并说明原因。

int fdrd, fdwr;

char c;

int main( int argc, char\* argv[] )

{

if ( argc != 3 )

{

exit( 1 );

}

if ( (fdrd=open(argv[1],O\_RDONLY)) == -1 )

{

exit( 1 );

}

if ( (fdwr=creat(argv[2],0666)) == -1 )

{

exit( 1 );

}

fork( );

rdwrt( );

exit( 0 );

}

void rdwrt( void )

{

for ( ; ; )

{

if ( read(fdrd,&c,1) != 1 )

{

return;

}

write( fdwrt, &c, 1 );

}

}

五、读程序，写出结果。

static void my\_exit1( void );

static void my\_exit2( void );

int main( void )

{

if ( atexit(my\_exit2) != 0 )

{

err\_sys( "can't register my\_exit2" );

}

if ( atexit(my\_exit1) != 0 )

{

err\_sys( "can't register my\_exit1" );

}

if ( atexit(my\_exit1) != 0 )

{

err\_sys( "can't register my\_exit1" );

}

printf( "main is done\n" );

return 0;

}

static void my\_exit1( void )

{

printf( "first exit handler\n" );

}

static void my\_exit2( void )

{

printf( "second exit handler\n" );

}

main is done

first exit handler

first exit handler

second exit handler

六、读程序，看程序是否有问题，问题并说明原因。如果有问题，请改正之。

static jmp\_buf env\_alrm;

int main( void )

{

int n;

char buf[MAXLINE];

if ( signal(SIGALRM,sig\_alrm) == SIG\_ERR )

{

exit( 1 );

}

alarm( 10 );

if ( setjmp(env\_alrm) != 0 )

{

printf( "read timeout" );

exit( 0 );

}

n = read( STDIN\_FILENO, buf, MAXLINE );

alarm( 0 );

write( STDOUT\_FILENO, buf, n );

exit( 0 );

}

static void sig\_alrm( int signo )

{

longjmp( env\_alrm, 1 );

}