

UC Day11

预习课

预习 内容

睡眠

睡眠

睡眠

- `#include <unistd.h>`
- `unsigned int sleep(unsigned int seconds);`
 - 功能：有限睡眠
 - 参数：seconds 以秒为单位的睡眠时限
 - 返回值：返回0或剩余秒数。
 - 该函数使调用进程睡眠seconds秒，除非有信号终止了调用进程或被其捕获
 - 如果有信号被调用进程捕获，在信号处理函数返回以后，sleep函数才会返回，且返回值为剩余的秒数，否则该函数将返回0，表示睡眠充足



睡眠

- `#include <unistd.h>`
- `int usleep (useconds_t usec);`
 - 功能：更精确的有限睡眠
 - 参数：usec 以微秒($1\text{微秒}=10^{-6}\text{秒}$)为单位的睡眠时限
 - 返回值：成功返回0,失败返回-1
 - 如果有信号被调用进程捕获，在信号处理函数返回以后，usleep函数才会返回，且返回值为-1，同时置errno为EINTR，表示阻塞的系统调用被信号中断



直播课见

UC

C/C++教学体系

目录

暂停

信号集

信号屏蔽

暂停



暂停

- `#include <unistd.h>`
- `int pause(void);`
 - 功能：无限睡眠
 - 返回值：成功阻塞，失败返回-1
 - 该函数使调用进(线)程进入无时限的睡眠状态，直到有信号终止了该进程或被其捕获。如果有信号被调用进程捕获，在信号处理函数返回以后，`pause`函数才会返回，其返回值-1，同时置`errno`为`EINTR`，表示阻塞的系统调用被信号打断。`pause`函数要么不返回，要么返回-1，永远不会返回0。



信号集

信号集

- 多个信号组成的信号集合谓之信号集
- 系统内核用sigset_t类型表示信号集
 - 在<signal.h>中又被定义为 `typedef __sigset_t sigset_t;`
 - 在<sigset.h>中有如下类型定义

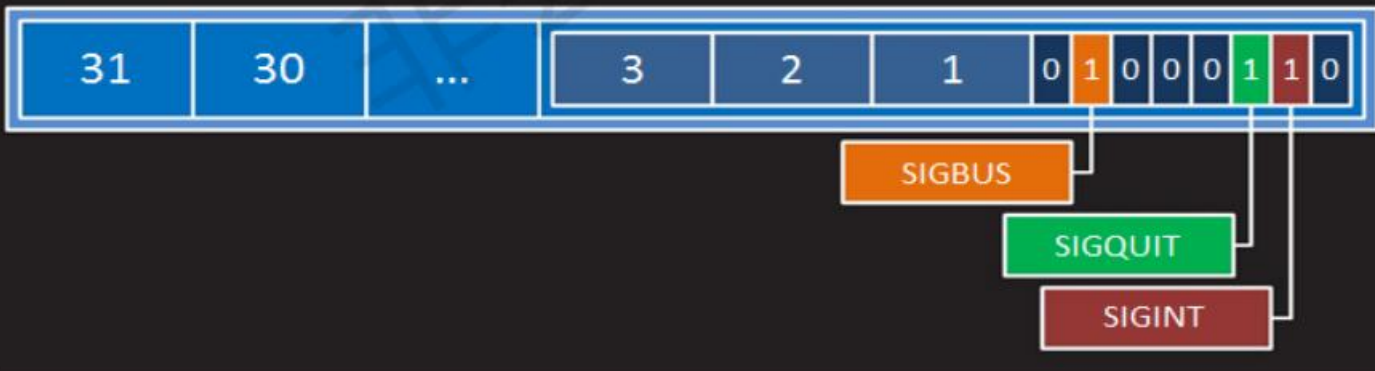
```
#define _SIGSET_NWORDS (1024 / (8 * sizeof (unsigned long int)))
typedef struct {
    unsigned long int __val[_SIGSET_NWORDS];
} __sigset_t;
```
- sigset_t类型是一个结构体，但该结构体中只有一个成员，是一个包含32个元素的整数数组（针对32位系统而言）



信号集

- 可以把sigset_t类型看成一个由1024个二进制位组成的大整数
 - 其中的每一位对应一个信号，其实目前远没有那么多信号
 - 某位为1就表示信号集中有此信号，反之为0就是无此信号
 - 当需要同时操作多个信号时，常以sigset_t作为函数的参数或返回值的类型

sigset_t



信号集

- `#include <signal.h>`
- `int sigfillset (sigset_t* sigset);`
 - 功能：填满信号集，即将信号集的全部信号位置1
 - 参数：sigset 信号集
 - 返回值：成功返回0，失败返回-1



信号集

- `#include <signal.h>`
- `int sigemptyset (sigset_t* sigset);`
 - 功能：清空信号集，即将信号集的全部信号位清0
 - 参数：sigset 信号集
 - 返回值：成功返回0，失败返回-1



信号集

- `#include <signal.h>`
- `int sigaddset (sigset_t* sigset, int signum);`
 - 功能：加入信号，即将信号集中与指定信号编号对应的信号位置1
 - 参数：sigset 信号集
 signum: 信号编号
 - 返回值：成功返回0，失败返回-1



信号集

- `#include <signal.h>`
- `int sigdelset (sigset_t* sigset, int signum);`
 - 功能：删除信号，即将信号集中与指定信号编号对应的信号位清0
 - 参数：sigset 信号集
 signum: 信号编号
 - 返回值：成功返回0，失败返回-1



信号集

- `#include <signal.h>`
- `int sigismember (const sigset_t* sigset, int signum);`
 - 功能：判断信号集中是否有某信号，即检查信号集中与指定信号编号对应的信号位是否为1
 - 参数：sigset 信号集
 signum: 信号编号
 - 返回值：有则返回1，没有返回0，失败返回-1



信号屏蔽

信号屏蔽

- 当信号产生时，系统内核会在其所维护的进程表中，为特定的进程设置一个与该信号相对应的标志位，这个过程就叫做递送(delivery)
- 信号从产生到完成递送之间存在一定的时间间隔，处于这段时间间隔中的信号状态称为未决(pending)
- 每个进程都有一个信号掩码(signal mask)，它实际上是一个信号集，位于该信号集中的信号一旦产生，并不会被递送给相应的进程，而是会被阻塞(block)在未决状态



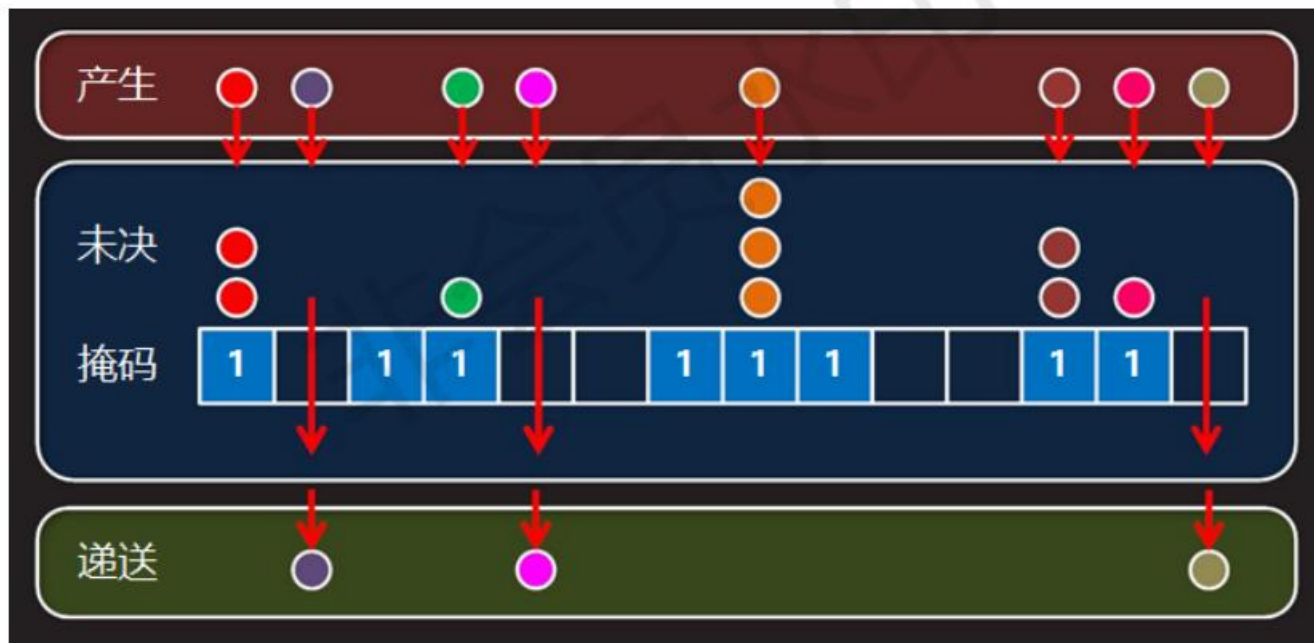
信号屏蔽

- 在信号处理函数执行期间，这个正在被处理的信号总是处于信号掩码中，如果又有该信号产生，则会被阻塞，直到上一个针对该信号的处理过程结束以后才会被递送
- 当进程正在执行类似更新数据库这样的敏感任务时，可能不希望被某些信号中断。这时可以通过信号掩码暂时屏蔽而非忽略掉这些信号，使其一旦产生即被阻塞于未决状态，待特定任务完成后，再回过头来处理这些信号



信号屏蔽

- 递送、未决与掩码



信号屏蔽

- `#include <signal.h>`
- `int sigprocmask (int how, const sigset_t* sigset, sigset_t* oldset);`
 - 功能：设置调用进程的信号掩码
 - 参数：how：修改信号掩码的方式，可取以下值
 - SIG_BLOCK - 将sigset中的信号加入当前信号掩码
 - SIG_UNBLOCK - 从当前信号掩码中删除sigset中的信号
 - SIG_SETMASK - 把sigset设置成当前信号掩码
 - sigset：信号集，取NULL则忽略此参数
 - oldset：输出原信号掩码，取NULL则忽略此参数
 - 返回值：成功返回0，失败返回-1



信号屏蔽

- `#include <signal.h>`
- `int sigpending (sigset_t* sigset);`
 - 功能：获取调用进程的未决信号集
 - 参数：sigset：输出未决信号集
 - 返回值：成功返回0，失败返回-1



信号屏蔽

- 对于可靠信号，通过sigprocmask函数设置信号掩码以后，每种被屏蔽信号中的每个信号都会被阻塞，并按先后顺序排队，一旦解除屏蔽，这些信号会被依次递送



信号屏蔽

- 对于不可靠信号，通过sigprocmask函数设置信号掩码以后，每种被屏蔽信号中只有第一个会被阻塞，并在解除屏蔽后被递送，其余的则全部丢失



谢谢

UC Day011

复习课

闹钟

非会员水印



闹钟

- `#include <unistd.h>`
- `unsigned int alarm(unsigned int seconds);`
 - 功能：设置闹钟
 - 参数：seconds 以秒为单位的闹钟时间。
 - 返回值：返回0或先前所设闹钟的剩余秒数。
 - alarm函数使系统内核在该函数被调用以后seconds秒的时候，向调用进程发送SIGALRM(14)信号
 - 若在调用该函数前已设过闹钟且尚未到期，则该函数会重设闹钟，并返回先前所设闹钟的剩余秒数，否则返回0
 - 若seconds取0，则表示取消先前设过且尚未到期的闹钟



下节课见