

从所有教程的词条中查询...

首页 > 慕课教程 > 物联网/嵌入式工程师 > 12.2 多路复用io-select(二) 应用

全部开发者教程

5.3 进程间通讯-信号量同步

6.1 线程的概念

7.1 线程相关命令

8.1 线程创建

8.2 线程的退出、等待与分离

8.3 创建多个线程

9.1 线程间通讯

10.1 线程互斥锁

10.2 线程同步

10.3 条件变量

11.1 阻塞 io 与非阻塞 io

12.1 多路复用io-select(一)-基本原理

12.2 多路复用io-select(二) 应用

12.3 多路复用io-select-底层



大白老师 · 更新于 2022-11-09

上一节 12.1 多路复用io... 12.3 多路复用io... 下一节

- select 多路复用io在实现时主要是以调用 **select** 函数来实现
- select 函数的基本信息如下:

函数头文件 #include <sys/select.h>

函数原型

```
int select(int nfds, fd_set *readfds, fd_set *writefds,
           fd_set *exceptfds, struct timeval *timeout);
```

函数功能 监控一组文件描述符，阻塞当前进程，由内核检测相应的文件描述符是否就绪,一旦有文件描述符就绪，将就绪的文件描述符拷贝给进程，唤醒进程处理

函数参数 nfds : 最大文件描述符加1

readfds : 读文件描述符集合的指针

writefds : 写文件描述符集合的指针

exceptfds : 其他文件描述符集合的指针

timeout : 超时时间结构体变量的指针

函数返回值

- 成功 : 返回已经就绪的文件描述符的个数
- 如果设置 timeout,超时就会返回 0
- 失败 : -1, 并设置 errno
- 操作文件描述符集合函数如下:

FD\_CLR

```
void FD_CLR(int fd,fd_set *set)
```

将 fd 从文件描述符集合中删除

fd : 文件描述符

set : 文件描述符集合的指针

FD\_ISSET

```
int FD_ISSET(int fd,fd_set *set)
```

判断 fd 是否在文件描述符集合中

fd : 文件描述符

意见反馈

收藏教程

标记书签

## FD\_SET

```
void FD_SET(int fd,fd_set *set)
```

将文件描述符添加到文件描述符集合中

fd : 文件描述符

set : 文件描述符集合的指针

## FD\_ZERO

```
void FD_ZERO(fd_set *set)
```

将文件描述符集合清空

set : 文件描述符集合的指针

## 示例

使用 select 函数监控标准输入, 如果有输入, 则打印相应的信息

<> 代码块

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/select.h>
4
5
6  int main(void)
7  {
8      int ret;
9      int maxfd = 0;
10     fd_set readfds,tmpfds;
11     struct timeval tv = {3,0},tmp_tv;
12     char buffer[64] = {0};
13
14     FD_ZERO(&readfds);
15     FD_SET(0,&readfds); // 将标准输入读文件描述符集合中
16
17     for(;;){
18         tmp_tv = tv;
19         tmpfds = readfds;
20
21         ret = select(maxfd + 1,&tmpfds,NULL,NULL,&tmp_tv);
22         if (ret == -1){
23
24             perror("[ERROR] select(): ");
25             exit(EXIT_FAILURE);
26
27         }else if (ret == 0){ // 超时返回
28
29             printf("Timeout.\n");
30
31         }else if (ret > 0){
32
33             if (FD_ISSET(0,&tmpfds)){ // 判断是否在集合中
34
35                 fgets(buffer,sizeof(buffer),stdin);
36                 printf("buffer : %s ",buffer);
37
38             }
39         }
40     }
41 }
```

意见反馈

收藏教程

标记书签

```
43         return 0;
    }
```

练习

使用 select 监听有名管道，当有名管道有数据时，读取数据并打印

 我要提出意见反馈

