

select poll epoll的区别

1 <https://www.cnblogs.com/Anker/p/3265058.html>

POLL函数

- 函数原型

```
1  int poll(struct pollfd *fds, nfds_t nfds, int timeout);
2
3  fds: 监听的文件描述符数组
4  struct pollfd {
5      int    fd;           /* file descriptor */ 监听的文件描述符
6      short  events;       /* requested events */ 待监听的文件描述符对应的监听事件: POL
7      short  revents;      /* returned events */ 传入时, 给0. 如果满足对应事件返回非0—
8  };
9  nfds: 监听数组的实际有效监听个数 (自己需要监听多少个)
10 timeout: 监听的时长: 以毫秒为单位
11 返回值: 满足对应监听事件的文件描述符的总个数
```

- 优点: 可以将监听事件集合和返回事件集合分离 (select在一块)
可以拓展监听上限 (超出1024)
- 缺点: 不能跨平台 (select可以)
无法直接定位满足监听事件的文件描述符 (和select一样)。编码难度较大

epoll: 函数

epoll时select和poll的增强

epoll使用一组函数来完成任务, epoll把用户关心的文件描述符上的事件放在内核中的一个事件表中, 从而无须想select和poll那样每次调用都要重新传入文件描述符或事件集。但是epoll需要**使用一个额外的文件描述符来唯一标识内核中的事件表**

1. int epoll_creat(int size): 返回值为一个epfd (fd, 下面可以挂子节点 (其他的文件描述符, 比如用于读的fd, 用于写的fd))

```
1  #include <sys/epoll.h>
2  int epoll_create(int size)
3  size参数现在还不起作用, 提示内核事件表需要多大
4  该函数返回的文件描述符将用作其他所有epoll系统调用的第一个参数, 以指定要访问的内核事件表
```

size表示红黑树的监听节点数量 (要挂的子节点的数量)

成功会返回一个非负的文件描述符

成功会返回红黑树的根节点，失败-1和errno

2. `epoll_ctl()`:这个函数用来操作epoll的内核事件表

```
1 int epoll_ctl(int epfd,int op,int fd,struct epoll_event *event)
2 fd参数是要操作的文件描述符，op参数指定操作类型。操作类型有如下三种
3 EPOLL_CTL_ADD;往事件表中注册fd的事件
4 EPOLL_CTL_MOD: 修改fd上的注册事件
5 EPOLL_CTL_DEL:删除fd上的注册事件
6 event参数指定事件，它是epoll_event结构体指针。epoll_event的定义如下
7 struct epoll_event
8 {
9     _uint32_t events;/epoll事件
10    epoll_data_t data;用户数据
11 };
12 其中event成员描述事件类型。epoll支持的事件类型和poll基本相同。表示epoll事件类型的宏是在poll对应
13 有两个额外的事件类型EPOLLET和EPOLLONESHOT.他们对于epoll的高效运作非常关键
14 data成员用于存储用户数据，其类型epoll_data_t的定义如下
15 typedef union epoll_data
16 {
17     void* ptr;
18     int fd;
19     uint32_t u32;
20     uint64_t u64;
21 }epoll_data_t;
```

3. `epoll_wait`函数

```
1 #include <sys/epoll.h>
2 int epoll_wait(int epfd,struct epoll_event* events,int maxevents,int timeout);
3 该函数成功时返回就绪的文件描述符的个数，失败时返回-1并设置errno。
4 epoll_wait函数如果监测到事件，就将所有就绪事件从内核事件表(由epfd参数指定)中复制到它的第二个参数
5 这个数组只用于输出epoll_wait检测到的就绪事件
```

4. 实现参考

1 <https://www.cnblogs.com/Anker/archive/2013/08/17/3263780.html>

5.

poll和epoll突破1024文件描述符设置

```
1 cat /proc/sys/fs/file-max
```

ulimit -a:查出来的是当前用户下的进程，默认打开文件描述符的最大个数。缺省为1024

```
1 sudo vi /etc/security/limits.config
2 在文件末尾写入以下配置，soft软限制，hard硬限制
3 * soft nofile 65535
4 * hard nofile 10000
```

- epoll:

6.