



- EPOLLIN: لقرءة. جاهزة ال بيانات
- EPOLLOUT: لكتابة. جاهز
- EPOLLERR: خطأ. حدث
- EPOLLHUP: لإتصال. إغلاق مثل الإتصال قطع

□ مثل أخرى بيانات أو موضح □ هو □ كما ملف مع لم struct epoll\_event في data حقل تخزن أن يمكن: ال مستخدم بيانات ال فعاليات. حدوث عند ال مصدر لتحدثي د مؤشر □

### 3. ال فعاليات ان تظار

تتبعه: يتم التي ال ملفات مع ال على ال فعاليات وان تظار لتوقف epoll\_wait استخدم

```
struct epoll_event events[MAX_EVENTS];
int nfd = epoll_wait(epoll_fd, events, MAX_EVENTS, -1);
```

□ ال معاملات:

- epoll\_fd: مثيل epoll.
- events: ال فعاليات لتخزين مصرفوفة
- MAX\_EVENTS: ال قصوى ال فعاليات عدد
- -1: فوراً. يعود 0 نهاية؛ بل ان تظار يعني -1 □ ثانية بالمللي ان تظار وقت: -1

□ ال خطأ. عند -1 أو □ nfd □ ال فعاليات على تحتوي ال ملفات مع ال عدد: تعود التي القيمة

### 4. ال فعاليات مع ال جة

وعمليته: epoll\_wait بواسطة إرجاعه تم التي ال فعاليات على عبر

```
for (int i = 0; i < nfd; i++) {
    if (events[i].events & EPOLLIN) {
        // `events[i].data.fd`
    }
}
```

□ ال فعالية. نوع لتحدثي □ events[i].events & EPOLLIN مثل ال من طقية ال معاملات باستخدام events حقل من تحقق ال فعالية. في تسبب الذي ال ملف مع لم لتحدثي events[i].data.fd استخدم

### 5. إخ تاري □ ال ملفات مع ال إدارة

□ ملف: مع لم تتبع لإيقاف EPOLL\_CTL\_DEL مع epoll\_ctl استخدم: إزالة

```
epoll_ctl(epoll_fd, EPOLL_CTL_DEL, some_fd, NULL);
```

□ تعديل: تعديل مع ال فعاليات تعديل: تعديل

```
ev.events = EPOLLOUT; //
epoll_ctl(epoll_fd, EPOLL_CTL_MOD, some_fd, &ev);
```

---

## رئىسي سية مفاهيم

حافة على تتبع مقابل مستوى على التتبع

- المقروءة. غير البيانات مثل الوضع استمر طالما متكرر بشكل epoll يبلغ: الالاف تراضي مستوى على التتبع الحالات. معظم في أسهل
- يتطلب جديدة. بيانات وصول مثل الوضع يتغير عندما واحدة مرة فقط يبلغ: EPOLLET حافة على التتبع تتعقيدًا. أكثر ولكن كفاءة أكثر الالفاعليات؛ فقدان لمنع EAGAIN حتى البيانات ج ميع قراءة كتابة
- الحافة. تتبع وضع تستخدم كنت إذا EPOLLIN | EPOLLET مثل ev.events بتحديث م

متزامن غير إدخال إخراج

وضع إلى مرفق بتحديث م الإدخال إخراج. عمليات على التوقف لمنع متزامن غير ملفات مع ال م غالبًا epoll يستخدم باسخدام: متزامن غير

```
fcntl(fd, F_SETFL, fcntl(fd, F_GETFL) | O_NONBLOCK);
```

---

## بسيط إكواد مثال:

من المستوى تتبع وضع يستخدم ال عملاء. إلى ال بيانات وإعادة الإتصالات لتق بيل epoll تستخدم لخدمة أساسى مثال هنا السهولة. أجل

```
#include <sys/epoll.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>

#define MAX_EVENTS 10
#define PORT 8080
```

```

int main() {
    //
    int listen_fd = socket(AF_INET, SOCK_STREAM, 0);
    if (listen_fd == -1) { perror("socket"); exit(1); }

    struct sockaddr_in addr = { .sin_family = AF_INET, .sin_addr.s_addr = INADDR_ANY, .sin_port = htons(PORT)
    if (bind(listen_fd, (struct sockaddr*)&addr, sizeof(addr)) == -1) { perror("bind"); exit(1); }
    if (listen(listen_fd, 5) == -1) { perror("listen"); exit(1); }

    //
    fcntl(listen_fd, F_SETFL, fcntl(listen_fd, F_GETFL) | O_NONBLOCK);

    //      `epoll`
    int epoll_fd = epoll_create1(0);
    if (epoll_fd == -1) { perror("epoll_create1"); exit(1); }

    //      `epoll`
    struct epoll_event ev, events[MAX_EVENTS];
    ev.events = EPOLLIN; //
    ev.data.fd = listen_fd;
    if (epoll_ctl(epoll_fd, EPOLL_CTL_ADD, listen_fd, &ev) == -1) { perror("epoll_ctl"); exit(1); }

    //
    while (1) {
        int nfds = epoll_wait(epoll_fd, events, MAX_EVENTS, -1);
        if (nfds == -1) { perror("epoll_wait"); exit(1); }

        for (int i = 0; i < nfds; i++) {
            int fd = events[i].data.fd;

            if (fd == listen_fd) {
                //
                int client_fd = accept(listen_fd, NULL, NULL);
                if (client_fd == -1) { perror("accept"); continue; }

                //
                fcntl(client_fd, F_SETFL, fcntl(client_fd, F_GETFL) | O_NONBLOCK);

                //      `epoll`
                ev.events = EPOLLIN;

```

```

    ev.data.fd = client_fd;
    if (epoll_ctl(epoll_fd, EPOLL_CTL_ADD, client_fd, &ev) == -1) {
        perror("epoll_ctl: client_fd");
        close(client_fd);
    }
} else {
    //
    char buf[1024];
    ssize_t n = read(fd, buf, sizeof(buf));
    if (n <= 0) {
        //          :          `epoll`
        if (n == -1 && (errno == EAGAIN || errno == EWOULDBLOCK)) continue;
        close(fd);
        epoll_ctl(epoll_fd, EPOLL_CTL_DEL, fd, NULL);
    } else {
        //
        write(fd, buf, n); // :
    }
}
}
}

close(epoll_fd);
close(listen_fd);
return 0;
}

```

## المثال على ملاحظات

- الجديدة. الإتصالات لتحديد EPOLLIN يتبع: الاستماع مرفق
- الواردة. البيانات لتحديد EPOLLIN أيضاً تتبع: العملاء مرفقات
- EPOLLOUT واستخدم الحافظة في البيانات بتخزين قم الإنتاج، في بالكامل. يكتمل write أن يفترض: تسوية الجزئية. للكتابة
- epoll من وإزالة الملف نهائية أو الأخطاء عند المرفقات يغلق: الأخطاء معالجة

## ملخص

تسجيل 2. epoll\_create باستخدام مثيل إنشاء 1. لينكس: في ملفات معالمة عدة مع لتعامل فعالة طريفة epoll يوفر في الفعاليات معالمة 4. epoll\_wait باستخدام الفعاليات ان تظار 3. epoll\_ctl باستخدام والفعاليات اللفات معالمة الحاجة. حسب المعالمة إزالة أو تتبعها يتم التي الفعاليات تعديل حلقة،

وضع استخدام في فكر الأداء، عالمة للاحتياجات بالنسبة الامستوى. تتبع وضع باستخدام يوصى البسيطة، لل تطبيقات أفضل على لل حصول متزامن غير إدخال وإخراج مع epoll استخدام دائماً بعناية. الامتاحة البيانات ج ميع معالمة مع الحافة تتبع اللنتائج.