gin 服务 无缝停启 更新服务

作者：ghqemperor
原文链接：https://ld246.com/article/1569057950169
来源网站：链滴

~~原文链接：https://juejin.im/post/5d2ed3776fb9a07edb397c77 (猜测)~~

- 在 kill 服务时，使用 kill -12 **** 将服务停止。
- 服务在收到信号量 12 (SIGUSR2) 后, 不再处理新请求，
- 将已开始的请求处理完成，
- 将 标准输出、错误输出 和 socket 的描述符转交给之后新启动的程序
- 无缝启停完成

例：kill -12 12345

```
package run

import (
    "context"
    "errors"
    "flag"
    "github.com/gin-gonic/gin"
    "log"
    "net"
    "net/http"
    "os"
    "os/exec"
    "os/signal"
    "syscall"
    "time"
)

var (
    listenerGin net.Listener = nil
    serverGin   *http.Server = nil
    gracefulGin              = flag.Bool("graceful", false, "listen on fd open 3(internal use only)")
)

func GinRun(router *gin.Engine, port string) {
    var err error
    flag.Parse()

    serverGin = &http.Server{
        Addr:    port,
        Handler: router,
    }
    // 判断是否为 reload
    if *gracefulGin {
        log.Println("listening on the existing file descriptor 3")
        f := os.NewFile(3, "")
        // 获取 上个服务程序的 socket 的描述符
        listenerGin, err = net.FileListener(f)
    } else {
        log.Println("listening on a new file descriptor")
        // 新建
        listenerGin, err = net.Listen("tcp", serverGin.Addr)
    }
```

```
    if err != nil {
        log.Printf("listener error: %v\n", err)
    }

    go func() {
        // 开启服务
        if err := serverGin.Serve(listenerGin); err != nil && err != http.ErrServerClosed {
            log.Printf("listen error:%v\n", err)
        }
    }()

    handleSig()
    log.Println("signal end")
}

func handleSig() {
    sign := make(chan os.Signal)
    signal.Notify(sign, syscall.SIGINT, syscall.SIGTERM, syscall.SIGUSR2)
    for {
        // 接收信号量
        sig := <-sign
        log.Printf("signal receive: %v\n", sig)
        ctx, _ := context.WithTimeout(context.Background(), time.Second*10)
        switch sig {
        case syscall.SIGINT, syscall.SIGTERM:
            // 关闭服务
            log.Println("shutdown")
            signal.Stop(sign)
            if err := serverGin.Shutdown(ctx); err != nil {
                log.Fatalf("[ service shutdown ] error:%v\n", err)
            }
            return
        case syscall.SIGUSR2:
            // 重启服务
            log.Println("reload")
            // 先启动新服务
            if err := reloadGin(); err != nil {
                log.Printf("[ service reload ] error: %v\n", err)
                continue
            }
            // 关闭旧服务
            if err := serverGin.Shutdown(ctx); err != nil {
                log.Fatalf("[ service reload ] shutdown error:%v\n", err)
            }
            log.Println("[ service reload ] success")
            return
        }
    }
}

func reloadGin() error {
    tl, ok := listenerGin.(*net.TCPListener)
    if !ok {
        return errors.New("listener is not tcp listener")
```

```
    }

    f, err := tl.File()
    if err != nil {
        return err
    }

    // 命令行启动新程序
    args := []string{"-graceful"}
    cmd := exec.Command(os.Args[0], args...)
    cmd.Stdout = os.Stdout         //  1
    cmd.Stderr = os.Stderr        //  2
    cmd.ExtraFiles = []*os.File{f} //  3

    return cmd.Start()
}
```

实例代码：demo.zip