

74.html”)

说明

--

go 中实现优雅的退出，主要使用了 `os/signal` 包，让程序能够接收到信号，并对信号执行一些操作。比如在linux下，也可以对一些信号进行捕捉，甚至改变信号所对应的行为。

退出部分的代码在`退出`标题里，其它代码只是我的一个测试环境，可以忽略。

这里我使用的是gin框架，并在gin框架中引入了os/signal包，实现对信号的处理（只做了部分信号的处理）。

并且有些信号不能被捕获，最常见的就是 `kill -9` 强杀，具体请看下最常见的信号列表。

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1d618b0dd386401090b76b7c29992ec1~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=690&h=706&s=144567&e=png&b=fafafa)

代码实现

...

```
func (p *router) Run(host, port string) {
//TODO 优化zap log日志，修改成自定义的zap 的logger

//err := p.engine.Ru nstant.ServerRestart*time.Second)
defer cancel()
if err := srv.Shutdown(ctx); err != nil {
logger.Fatalf("Server Shutdown: ", err)
}
select {
case <-ctx.Done():
```

```
}  
logger.Info("Server exiting...")  
}  
  
func Recover() {  
if p := recover(); p != nil {  
fmt.Println("p = ", p)  
}  
}  
  
...
```

如下，捕捉到了ctrl+c对应的信号之后，执行了退出操作，重启服务的时候也可以执行退出操作。

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/a4b460e6002b45bdbdc48dfc368c11ac~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=988&h=83&s=18966&e=png&b=000000)

原文链接: <https://juejin.cn/post/7352075798068412442>