链滴

优雅 panic

作者：opsxdev

原文链接：https://ld246.com/article/1616038330992

来源网站：链滴

优雅panic

1.预期宕机和非预期宕机

go

```
func forEachNode(n *html.Node, pre, post func(n *html.Node)) {
    if pre != nil {
        pre(n)
    }
    for c := n.FirstChild; c != nil; c = c.NextSibling {
        forEachNode(c, pre, post)
    }
    if post != nil {
        post(n)
    }
}

//!+
// soleTitle returns the text of the first non-empty title element
// in doc, and an error if there was not exactly one.
func soleTitle(doc *html.Node) (title string, err error) {
    type bailout struct{}

    defer func() {
        switch p := recover(); p {
        case nil:
            // no panic
        case bailout{}:
            // "expected" panic
```

```
            err = fmt.Errorf("multiple title elements")
        default:
            panic(p) // unexpected panic; carry on panicking
        }
    }()

    // Bail out of recursion if we find more than one non-empty title.
    forEachNode(doc, func(n *html.Node) {
        if n.Type == html.ElementNode && n.Data == "title" &&
            n.FirstChild != nil {
            if title != "" {
                panic(bailout{}) // multiple title elements
            }
            title = n.FirstChild.Data
        }
    }, nil)
    if title == "" {
        return "", fmt.Errorf("no title element")
    }
    return title, nil
}

//!-

func title(url string) error {
    resp, err := http.Get(url)
    if err != nil {
        return err
    }

    // Check Content-Type is HTML (e.g., "text/html; charset=utf-8").
    ct := resp.Header.Get("Content-Type")
    if ct != "text/html" && !strings.HasPrefix(ct, "text/html;") {
        resp.Body.Close()
        return fmt.Errorf("%s has type %s, not text/html", url, ct)
    }

    doc, err := html.Parse(resp.Body)
    resp.Body.Close()
    if err != nil {
        return fmt.Errorf("parsing %s as HTML: %v", url, err)
    }
    title, err := soleTitle(doc)
    if err != nil {
        return err
    }
    fmt.Println(title)
    return nil
}

func main() {
    for _, arg := range os.Args[1:] {
        if err := title(arg); err != nil {
            fmt.Fprintf(os.Stderr, "title: %v\n", err)
```

```
        }
    }
}
```

go