

Escrevendo em múltiplas linhas

Transcrição

Agora que conseguimos criar e abrir o arquivo na função `registraLog`, não podemos nos esquecer de fechá-lo

```
// restante do código omitido

func registraLog(site string, status bool) {

    arquivo, err := os.OpenFile("log.txt", os.O_CREATE|os.O_RDWR, 0666)

    if err != nil {
        fmt.Println("Ocorreu um erro:", err)
    }

    fmt.Println(arquivo)

    arquivo.Close()
}
```

Agora, ao invés de imprimir o ponteiro para o arquivo, queremos escrever nele. Vamos começar escrevendo somente se o site carregou ou não.

Escrevendo no arquivo

Para escrever no arquivo, ele possui a função `WriteString`, ou seja, basta passar o texto para essa função, que ela o escreve dentro do arquivo:

```
// restante do código omitido

func registraLog(site string, status bool) {

    arquivo, err := os.OpenFile("log.txt", os.O_CREATE|os.O_RDWR, 0666)

    if err != nil {
        fmt.Println("Ocorreu um erro:", err)
    }

    arquivo.WriteString(site + " - online: " + status + "\n")

    arquivo.Close()
}
```

Mas o Go irá reclamar, isso acontece pois a função `WriteString` recebe uma string, e nós estamos tentando concatenar com um booleano, então precisamos convertê-lo com a função `strings`. Para converter um tipo booleano em uma string, vamos utilizar o pacote `strconv`, que possui a função `FormatBool`:

```
// restante do código omitido
```

```
func registraLog(site string, status bool) {  
  
    arquivo, err := os.OpenFile("log.txt", os.O_CREATE|os.O_RDWR, 0666)  
  
    if err != nil {  
        fmt.Println("Ocorreu um erro:", err)  
    }  
  
    arquivo.WriteString(site + " - online: " + strconv.FormatBool(status) + "\n")  
  
    arquivo.Close()  
}
```

Ao iniciar o monitoramento e ficar verificando o arquivo de logs, vemos que as linhas são sobrescritas a cada site novo monitorado. Isso acontece porque por padrão, quando escrevemos em um arquivo, é sempre escrito do seu começo. Para isso, na hora de abrir o arquivo, existe a *flag* `O_APPEND`, que faz com que o texto seja escrito ao final do arquivo:

```
// restante do código omitido  
  
func registraLog(site string, status bool) {  
  
    arquivo, err := os.OpenFile("log.txt", os.O_CREATE|os.O_RDWR|os.O_APPEND, 0666)  
  
    if err != nil {  
        fmt.Println("Ocorreu um erro:", err)  
    }  
  
    arquivo.WriteString(site + " - online: " + strconv.FormatBool(status) + "\n")  
  
    arquivo.Close()  
}
```

Agora conseguimos imprimir corretamente no arquivo, faltando escrever o horário em que o site estava online ou offline. Faremos isso no próximo vídeo :)