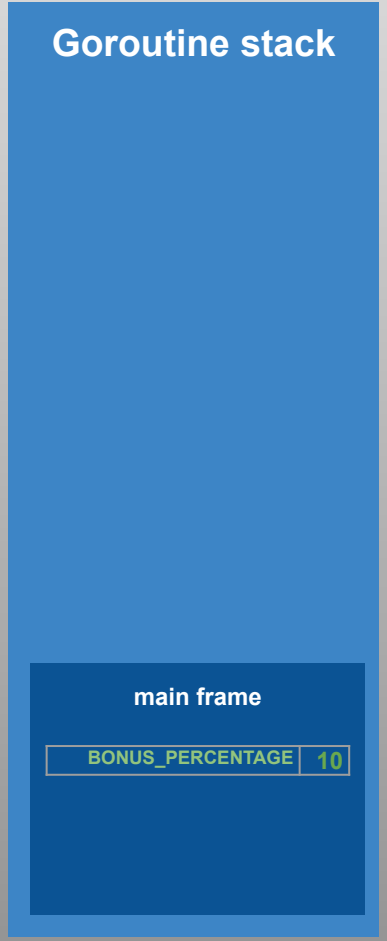
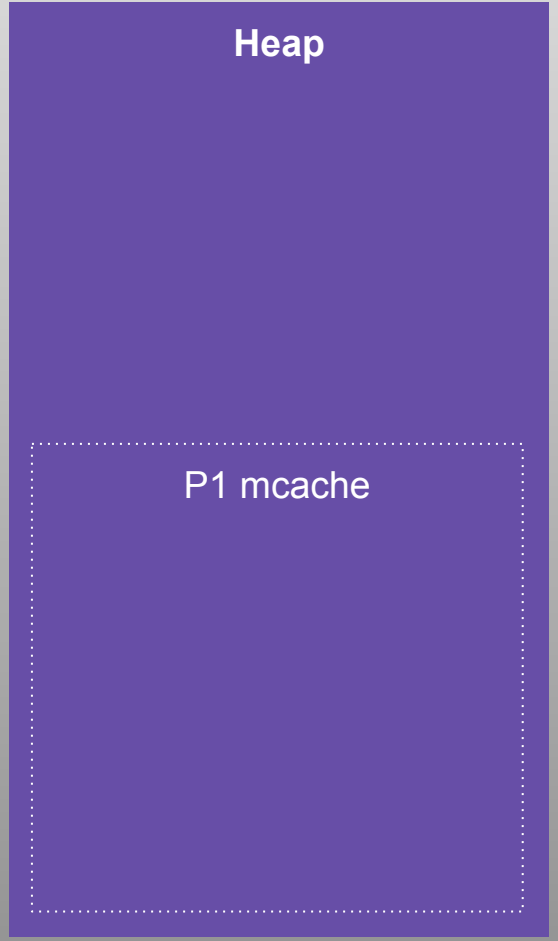
Go Memory usage

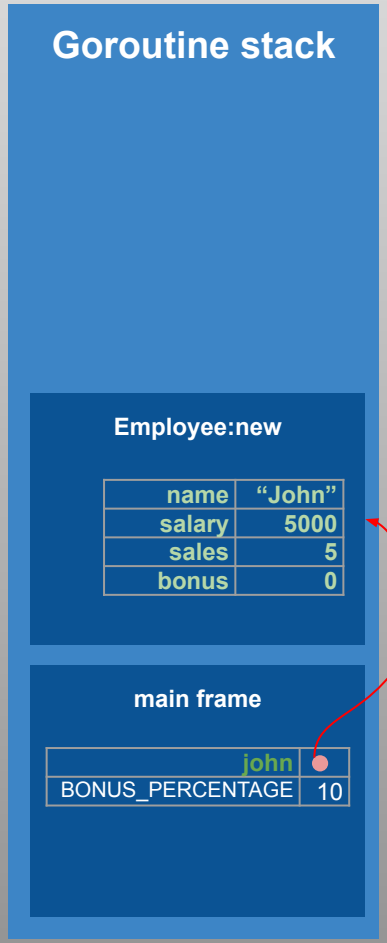
```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

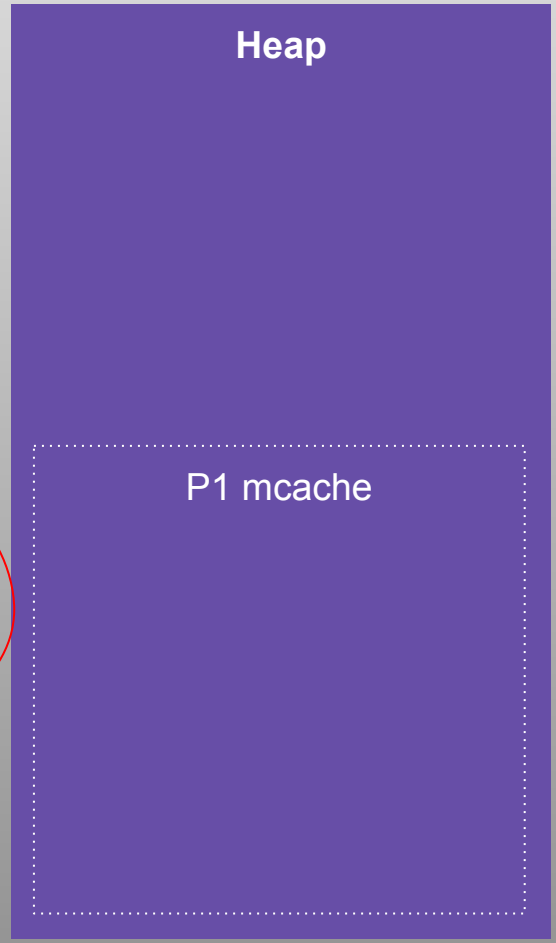




Go Memory usage

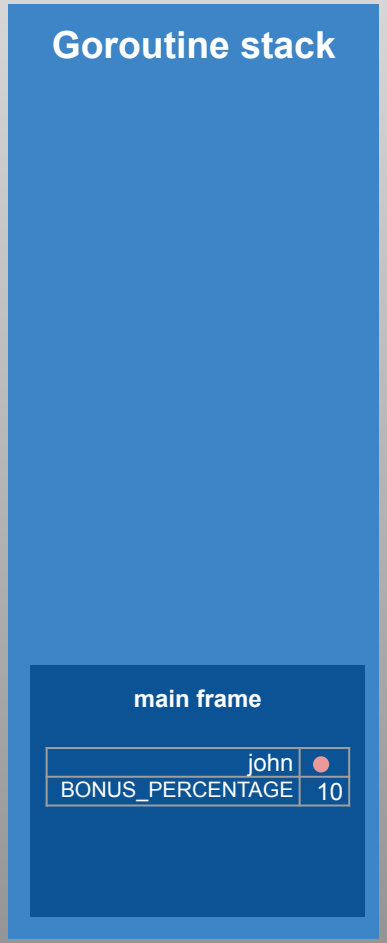
```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

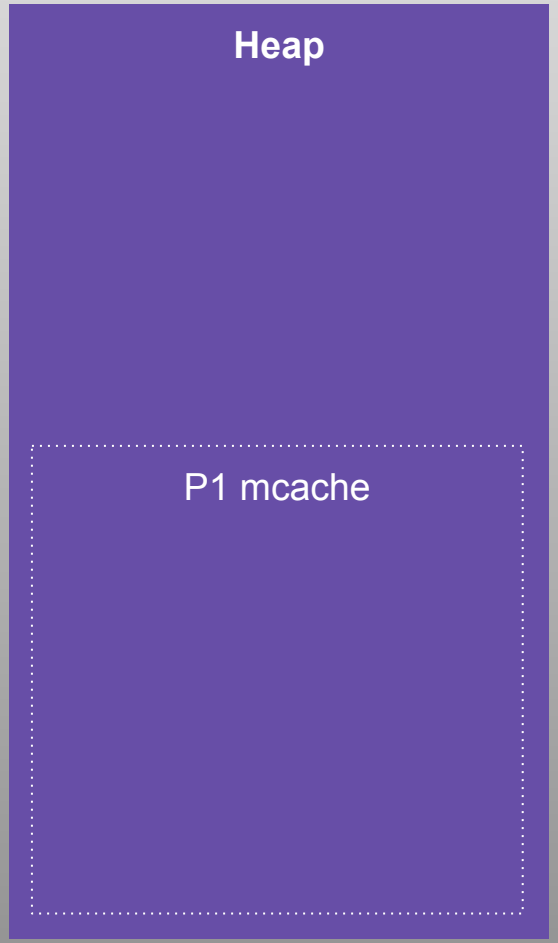




Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```





Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

Goroutine stack

findEmployeeBonus

salary	5000
noOfSales	5

main frame

john	●
BONUS_PERCENTAGE	10

Heap

P1 mcache

Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

Goroutine stack

findEmployeeBonus

salary	5000
noOfSales	5
bonusPercentage	0

main frame

john	●
BONUS_PERCENTAGE	10

Heap

P1 mcache

Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

Goroutine stack

getBonusPercentage

salary	5000

findEmployeeBonus

salary	5000
noOfSales	5
bonusPercentage	0

main frame

john	●
BONUS_PERCENTAGE	10

Heap

P1 mcache

Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

Goroutine stack

getBonusPercentage

salary	5000
percentage	500

findEmployeeBonus

salary	5000
noOfSales	5
bonusPercentage	0

main frame

john	●
BONUS_PERCENTAGE	10

Heap

P1 mcache

Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

Goroutine stack

getBonusPercentage

salary	5000
percentage	500
return	500

findEmployeeBonus

salary	5000
noOfSales	5
bonusPercentage	0

main frame

john	●
BONUS_PERCENTAGE	10

Heap

P1 mcache

Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

Goroutine stack

findEmployeeBonus

salary	5000
noOfSales	5
bonusPercentage	500

main frame

john	●
BONUS_PERCENTAGE	10

Heap

P1 mcache

Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

Goroutine stack

findEmployeeBonus

salary	5000
noOfSales	5
bonusPercentage	500
bonus	2500

main frame

john	●
BONUS_PERCENTAGE	10

Heap

P1 mcache

Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```

Goroutine stack

findEmployeeBonus

salary	5000
noOfSales	5
bonusPercentage	500
bonus	2500
return	2500

main frame

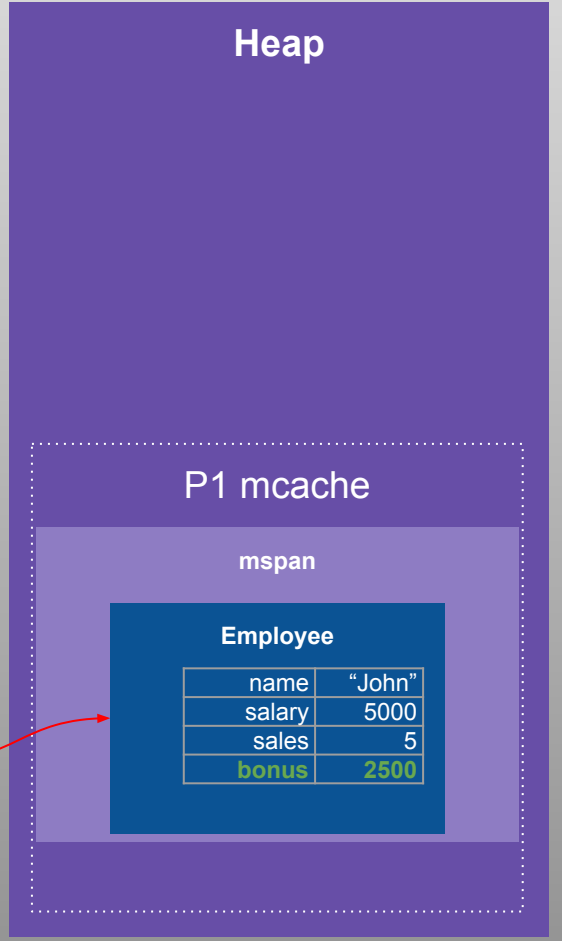
john	●
BONUS_PERCENTAGE	10

Heap

P1 mcache

Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```



Go Memory usage

```go
package main

import (
    "fmt"
)

type Employee struct {
    name   string
    salary int
    sales  int
    bonus  int
}

const BONUS_PERCENTAGE = 10

func getBonusPercentage(salary int) int {
    percentage := (salary * BONUS_PERCENTAGE) / 100
    return percentage
}

func findEmployeeBonus(salary, noOfSales int) int {
    bonusPercentage := getBonusPercentage(salary)
    bonus := bonusPercentage * noOfSales
    return bonus
}

func main() {
    var john = Employee{"John", 5000, 5, 0}
    john.bonus = findEmployeeBonus(john.salary, john.sales)
    fmt.Println(john.bonus)
}
```