PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL – ESCOLA POLITÉCNICA
Fundamentos de Processamento Paralelo e Distribuído - Prof. Fernando Dotti

---

Exercícios: Concorrência e Canais

No código abaixo considere que cada atribuição é atômica (indivisível). Considere como estado do sistema as configurações possíveis da dupla de variáveis globais [x, y]. O estado inicial é [0, 0]. Responda quais diagramas de estados e transições representam a execução de:

a) questaoStSp1()
b) questaoStSp2()
c) questaoStSp3()
d) questaoStSp4()

```
var x, y int = 0, 0
func px() {
      x = 1
      x = 2
}
func py() {
      y = 1
      y = 2
}
func questaoStSp1() {
      px()
      py()
}
func questaoStSp2() {
      go px()
      py()
}
```

```
func pxs(c chan int) {
      x = <- c
      x = 2
}
func pys(c chan int) {
      y = 1
      c <- y
      y = 2
}
func questaoStSp3() {
      c := make(chan int, 0)
      go pxs(c)
      go pys(c)
}
```

```
func pxs2(c chan int) {
      x = 1
      <- c
      x = 2
}
func pys2(c chan int) {
      y = 1
      c <- y
      y = 2
}
func questaoStSp4() {
      c := make(chan int, 0)
      go pxs2(c)
      go pys2(c)
}
```

Para questões envolvendo canais de comunicação considere, além da linguagem Go, a seguinte sintaxe para algoritmos:

c = cria canal [tipo, tam]
// c é um canal com buffer de tamanho tam que armazena elementos de tipo
Var x := <- c // x recebe o valor lido de c
c <- y // escreve o valor y em c

1. Com relação aos programas abaixo:
   a. escreva as saídas possíveis geradas pelo programa 1 (obs.: “descrever” é dizer como são; e não quais são), considerando N=0
   b. Existe diferença se N > 0 ? Justifique.
   c. Todos os valores de ambos processos serão lidos em t3 ? Justifique
   d. Existe diferença entre as saídas do Programa1 e do Programa2? Justifique

Programa 1

```
Constante N
Vaiáveis globais canal c
```

```
Thread t1 {
 Int i=1
 Loop {
    c<-i
    i:=i+2
   }
}
```

```
Thread t2 {
 Int i=2
 Loop {
    c<-i
    i:=i+2
   }
}
```

```
Thread t3 {
  Loop {
    Print(<-c, " , ")
  }
}
```

```
Main {
  c = cria canal(int,N)
  Inicia t1, t2 e t3
  Espera término de t1, t2 e t3
}
```

Programa 2

```
Constante N
Vaiáveis globais canal c1 , c2

Thread t1 {
 Int i=1
 Loop {
    c1<-i
    i:=i+2
   }
}

Thread t2 {
 Int i=2
 Loop {
    c2<-i
    i:=i+2
   }
}

Thread t3 {
  Int x
  Loop {
    Select {
     Case: x<-c1
           break
     Case: x<-c2
           break
     Print(<-c, " , ")
  }
}

Main {
  c1 = cria canal(int,N)
  c2 = cria canal(int,N)
  Inicia t1, t2 e t3
  Espera término de t1, t2 e t3
}
```

Dados os programas abaixo, responda as perguntas colocadas nos comentários (diretivas package e imports omitidos por espaço):

```
func qA(i int) {
      fmt.Println(i)
}
func questaoA() {
      for i := 0; i < 10; i++ {   go qA(i)   }
      // Q.A.1  qual o numero de processos ativos neste ponto ?
      //        resposta pode ser um intervalo.         R: _____________________________
}
func main() {  questaoA()  }
      // Q.A.2  como podem ser as possíveis saidas de "questaoA()"
      // descreva como elas podem ser, e não exemplos concretos
```

```
func qB(i int, c chan struct{}) {
      fmt.Println(i)
      c <- struct{}{}
}
func questaoB() {
      c := make(chan struct{})
      for i := 0; i < 10; i++ {  go qB(i, c)   }
      // Q.B.1  qual o numero de processos ativos neste ponto ? R: ___________________
      for i := 0; i < 10; i++ {  <-c  }
      // Q.B.2 qual o numero de processos ativos neste ponto ?  R: ___________________
}
func main() {  questaoB()  }
      // Q.B.3  como podem ser as possíveis saidas de "questaoB()"
      // descreva como elas podem ser, e não exemplos concretos
```

```
const X = 40
var ch [X]chan struct{}
func fA(id int, in chan struct{}, out chan struct{}) {
	for {
		<-in
		fmt.Println(id)
		out <- struct{}{}
	}
}
func questaoC() {
	for i := 0; i < X; i++ {
		ch[i] = make(chan struct{})
	}
	for i := 0; i < X; i++ {
		go fA(i, ch[i], ch[(i+1)%X])
	}
	ch[0] <- struct{}{}
	blq := make(chan struct{}) // bloqueia
	<-blq
}
// Q.C.1 descreva como é a saída gerada pela execução de questaoC()
```

```
func gera(c chan string, s string) {
	for {   c <- s   }
}
func questaoD() {
	c := make(chan string)
	go gera(c, "a")
	go gera(c, "b")
	for {   fmt.Print(<-c)   }
}
func main() {   questaoD()  }

// Q.D.1 marque a(s) resposta(s) possível(is) para o comportamento da "questaoD()"
// a) bloqueio
// b) prints de sequencias de ab's
// c) prints de valores corrompidos por escrita concorrente
// d) prints de qualquer sequencia composta por a's e b's,
//    ou somente por a’s ou somente por b’s
// e) prints de qualquer sequencia composta por a's e b's

// Q.D.2  justifique cada resposta marcada  (use a folha de respostas)
```

```
const N = 0   // ou N = 1
func escreveLe(v int, e chan int, l chan int, fin chan struct{}) {
      for i := 0; i < 1000; i++ {
            e <- v
            <-l
      }
      fin <- struct{}{}
}
func questaoE() {
      c1 := make(chan int, N)
      c2 := make(chan int, N)
      chfin := make(chan struct{})
      go escreveLe(1, c1, c2, chfin)
      go escreveLe(2, c2, c1, chfin)
      <-chfin
      <-chfin
}
func main() {   questaoE() }
// Q.E.1  existe diferença de comportamento no caso de N=0 e N=1 ?
//        justifique sua resposta.
```

```
func qR2(i int) {
      go qR1(i - 1)
}
func qR1(i int) {
      if i > 0 {
            go qR2(i)
            fmt.Println(i)
      }
}
func questaoR() {
      go qR1(5)
}
// Q.R.1 quantos processos ao todo a execução de questaoR() cria ?
// Q.R.2 descreva como podem ser as saídas possíveis de questaoR()
```

```
func qS(c chan int, n int) {
      for i := 1; i < n; i++ {
            if rand.Intn(100) >= 50 {
                  c <- i
            } else {
                  fmt.Println(<-c)
            }
      }
}
func questaoS() {
      c := make(chan int,0)
      rand.Seed(86)  // ou qualquer valor
      go qS(c, 3)
      go qS(c, 3)
      <-make(chan struct{})
}

// Q.questaoS.1   quais as possíveis saídas deste programa ?
//                justifique as respostas
```