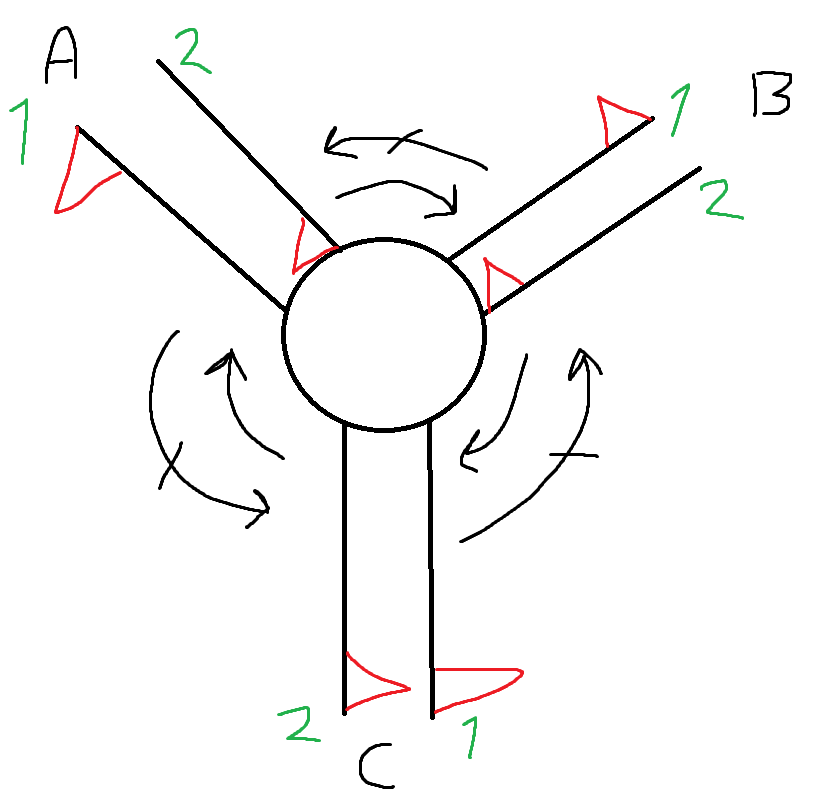
# Three process Kassel’s

Demonstrated by Ýmir Þórleifsson & Bjartur Þórhallsson.

To solve this problem, we simply need to add a new flag for each dog owner and change the logic behind raising flags when a dog owner wants to let their dog outside and the checks for if it's safe for a dog owner to put their dog in the garden.

The first flag keeps its purpose of being the flag that signals if a dog owner wants to put their dog in the garden. Then next two flags are a little more complex, see the image below (does not depict flag 0, only 1 and 2 for each dog owner):

Simply put:  
Alice makes her flag1 the opposite of Charlie’s flag2, and her flag2 the same as Bob’s flag1.  
Bob makes his flag1 the opposite of Alice’s flag2, and his flag2 the same as Charlie’s flag1.  
Charlie makes his flag1 the opposite of Bob’s flag2, and his flag2 the same as Alice’s flag1.

This way we can keep the same logic as when we had two dog owners; the first one to request access to the garden is always ahead of the next one, then one after that knows he is ahead of the third and the last one knows he is last, until the first one lets his dog inside and then he knows he is ahead of that dog owner.

The next step was to make some additions to the conditional statements in the loop. The idea behind the changes is the same as for Kessel’s algorithm, except it’s been refurbished to support three dog owners.

Here is a working code snippet of our solution:

|  |
| --- |
| *package* main  *import* "fmt"  *var* bobs\_flag [3]bool  *var* alices\_flag [3]bool  *var* charlies\_flag [3]bool  *func* *Alice*() {  *for* {  // *Remainder of code*  alices\_flag[0] = true  alices\_flag[1] = !charlies\_flag[2]  alices\_flag[2] = bobs\_flag[1]  loop := true  *for* loop {  // Both other *flags are up*  *if* bobs\_flag[0] == true && charlies\_flag[0] == true {  *if* alices\_flag[1] == charlies\_flag[2] && alices\_flag[2] != bobs\_flag[1] {  loop = false  }  }  // *Bob’s flag is up*  *if* bobs\_flag[0] == true && charlies\_flag[0] == false {  *if* alices\_flag[2] != bobs\_flag[1] {  loop = false  }  }  // *Charlie’s flag is up*  *if* charlies\_flag[0] == true && bobs\_flag[0] == false {  *if* alices\_flag[1] == charlies\_flag[2] {  loop = false  }  }  // *No flags are up*  *if* charlies\_flag[0] == false && bobs\_flag[0] == false {  loop = false  }  }  fmt.Println("Alice is entering critical section")  // *Critical section*  fmt.Println("Alice has exited critical section")  alices\_flag[0] = false  }  }    *func* *Bob*() {  *for* {  // *Remainder of code*  bobs\_flag[0] = true  bobs\_flag[1] = !alices\_flag[2]  bobs\_flag[2] = charlies\_flag[1]  loop := true  *for* loop {  // Both other *flags are up*  *if* charlies\_flag[0] == true && alices\_flag[0] == true {  *if* bobs\_flag[1] == alices\_flag[2] && bobs\_flag[2] != charlies\_flag[1] {  loop = false  }  }  // *Charlie's flag is up*  *if* charlies\_flag[0] == true && alices\_flag[0] == false {  *if* bobs\_flag[2] != charlies\_flag[1] {  loop = false  }  }  // *Alices flag is up*  *if* alices\_flag[0] == true && charlies\_flag[0] == false {  *if* bobs\_flag[1] == alices\_flag[2] {  loop = false  }  }  // *No flags are up*  *if* bobs\_flag[0] == false && alices\_flag[0] == false {  loop = false  }  }  fmt.Println("Bob is entering critical section")  // *Critical section*  fmt.Println("Bob has exited critical section")  bobs\_flag[0] = false  }  }    *func* *Charlie*() {  *for* {  // *Remainder of code*  charlies\_flag[0] = true  charlies\_flag[1] = !bobs\_flag[2]  charlies\_flag[2] = alices\_flag[1]  loop := true  *for* loop {  // Both other *flags are up*  *if* bobs\_flag[0] == true && alices\_flag[0] == true {  *if* charlies\_flag[1] == bobs\_flag[2] && charlies\_flag[2] != alices\_flag[1] {  loop = false  }  }  // *Bob's flag is up*  *if* bobs\_flag[0] == true && alices\_flag[0] == false {  *if* charlies\_flag[1] == bobs\_flag[2] {  loop = false  }  }  // *Alices flag is up*  *if* alices\_flag[0] == true && bobs\_flag[0] == false {  *if* charlies\_flag[2] != alices\_flag[1] {  loop = false  }  }  // *No flags are up*  *if* bobs\_flag[0] == false && alices\_flag[0] == false {  loop = false  }  }  fmt.Println("Charlie is entering critical section")  // *Critical section*  fmt.Println("Charlie has exited critical section")  charlies\_flag[0] = false  }  }    *func* *main*() {  *go* Alice()  *go* Bob()  Charlie()  } |