The Bakery Algorithm

```
--algorithm Bakery
{ variable num = [i ∈ Procs ↦ 0], flag = [i ∈ Procs ↦ FALSE] ;
  process ( p ∈ Procs )
    variables unchecked = {}, max = 0, nxt = 1 ;
    { ncs: while ( TRUE )
           { e1:  either { flag[self] := ¬flag[self] ;
                           goto e1 }
                  or     { flag[self] := TRUE ;
                           unchecked := Procs \ {self} ;
                           max := 0
                         } ;

             e2:  while ( unchecked ≠ {} )
                    { with ( i ∈ unchecked )
                        { unchecked := unchecked \ {i} ;
                          if ( num[i] > max ) { max := num[i] }
                        }
                    } ;
             e3:  either { with ( k ∈ Nat ) { num[self] := k } ;
                           goto e3 }
                  or     { with ( i ∈ {j ∈ Nat : j > max} )
                             { num[self] := i }
                         } ;
             e4:  either { flag[self] := ¬flag[self] ;
                           goto e4 }
                  or     { flag[self] := FALSE ;
                             unchecked := Procs \ {self}
                         } ;
             w1:  while ( unchecked ≠ {} )
                    {    with ( i ∈ unchecked ) { nxt := i } ;
                         await ¬flag[nxt] ;
                    w2:  await  ∨ num[nxt] = 0
                                ∨ ⟨num[self], self⟩ ≺ ⟨num[nxt], nxt⟩ ;
                         unchecked := unchecked \ {nxt} ;
                    } ;
             cs:  skip ;    the critical section;
             exit: either { with ( k ∈ Nat ) { num[self] := k } ;
                            goto exit }
                   or     { num[self] := 0 }
           }
    }
}
```

CLOSE