

# Le dîner des philosophes

# Le problème

- 5 philosophes sont autour d'une table ronde ;
- Le dîner se résume à manger et à penser, le tout en boucle ;
- Pour manger, un philosophe doit prendre deux baguettes (une à sa droite et une à sa gauche) ;
- Il n'y a que 5 baguettes (une entre chaque philosophe)

# Que fait chaque philosophe ?

- Un philosophe commence à penser,
- lorsqu'il a faim, il essaye de prendre les baguettes qui sont immédiatement à côté de lui (pas d'autres),
- il a besoin de deux baguettes pour manger et ne peut prendre une baguette qui est dans la main d'un voisin,
- Quand il a ses deux baguettes dans les mains, il mange sans les libérer. Dans le cas contraire, il doit attendre à nouveau qu'elles deviennent libres,
- Quand il a fini de manger, il repose les deux baguettes et recommence à penser.

# Un début de schéma algorithmique

```
while (1) {  
    prendre la baguette de droite  
    prendre la baguette de gauche  
    manger();  
    libérer la baguette de droite  
    libérer la baguette de gauche  
    penser();  
}
```

Le but est de résoudre le problème de synchronisation sous-jacent +

- tout le monde doit pouvoir manger
- limiter les attentes (inutiles et inefficaces)