- No preemption
- No OS support
- Problems so on

- Critical Section problem
- 2 processes
- Wait
- Mutual
- Exclusion
- Bounded Waiting
- Using Software
- Signal
Process\[0\] = 1;

turn = 1;

while (flag[1] == 1 && turn == 1)

{
  turn = 1;
  flag[0] = 1;
  Process[0]
}

---------

Critical Section

---------

more than 2 processes

as N processes

Dawok Balent

deGriff

ἔριθεν

διαλύσεων
More evidence is needed than the evidence as shown on a number sequence. When evidence and person
to process

new_array

\[ \text{new} \left[ i \right] = \text{new} \left[ i \right] + \text{new} \left[ i \right] \]

new

\[ \text{new} \left[ i \right] = 0 \Rightarrow \text{new} \left[ i \right] \]

num \[ \text{new} \left[ \right] \] of picked

num \[ \text{new} \left[ \right] \] of array of picked
\[
\text{num}_{\mathbb{N}} = \max\{\text{num}(0), \ldots, \text{num}(n)\} + 1
\]

Please fix a num:
Bakery Algorithm (1974)
Proc \([i]\)

```
choosing\[i\] = 1;
num[0] = max(num[0],num[1]...num[n-1])+1;
choosing\[i\] = 0;
for j = 0 to n-1 {
    while (choosing[j] == 1) <no-op>;
    while (num[j] <> 0 and (num[j],j) < (num[i],i) <no-op>;
    num[j] = max(num[0],num[1]...num[n-1])+1;
    critical section
};
```

Critical Section
Therefore, if $x > y$, then

$[x, y] = [\min(x, y), \max(x, y)]$.

But if $x < y$, then $y < x$.

Thus, if $x > y$, then $y < x$.

($\max(x, y), y$) > ($\min(x, y), x$).