

Musterlösung coarsegrained Bakery beschränken

```
int turn[1:n] = ([n] 0);

process CS[i = 1 to n] {
  while (true) {
    <await (max(turn[1:n]) < MAXINT/2)>
    <turn[i] = max(turn[1:n]) + 1>
    for [j = 1 to n st j != i]
      <await (turn[j] == 0 or turn[i] < turn[j]);>
    CS;
    turn[i] = 0;
    NCS;
  }
}
```

After the maximum turn reaches the constant $\text{MAXINT}/2$ all processes are delaying their execution in the delay statement until all processes with $\text{turn} > \text{MAXINT}/2$ have completed their CS and reset their turn to 0. As the await statement is assumed to behave as a FIFO the waiting processes are then released in order of arrival, keeping the same fairness as the unmodified algorithm. Unnecessary delay is avoided as no other process trying to enter implies $\text{max}(\text{turn}[1:n]) == 0 < \text{MAXINT}/2$ and thus no waiting.

Mutual exclusion is untouched as the entry protocol is unchanged after the additional await statement.