reach A

reach B

must have A or B

2

for synchronization

non preemptive

Semaphore

Process 2
Let's count to 8.

```python
if a certain condition is satisfied:
    proceed/therefore
donor + something
```
ind-Sam (Sam, value)

Sam . count = value

Sam . G = empty

get-Sam
Analysis

- Boat (ocean front)

- Computer software (PIE)

- Dog pneumonia

- Parental application
as threads of others run application
as core to memory, mig
- Master worker kernels
- function run on all cores
- symmetric kernel (no kernel)

Kernel Structure
C  passes through & interacts with system A

\[ \text{Blocks} \]

- Blacklisted (Background)
Hello

Send RCU system

Read

Kernel

not a user

Apple process

0 0 0

mucalmean access feature

OS process (apple)
Windows = microkernel separated OS & kernel

\( \text{Linux} \) \rightarrow \text{kernel} \rightarrow \text{Monolithic kernel}
Theorem

After some help, the source of 
the example is

\[ \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{2}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

\( \Rightarrow \)

CPU

(assume symmetric)

\[ \cdots \]

multi-processor scheduling