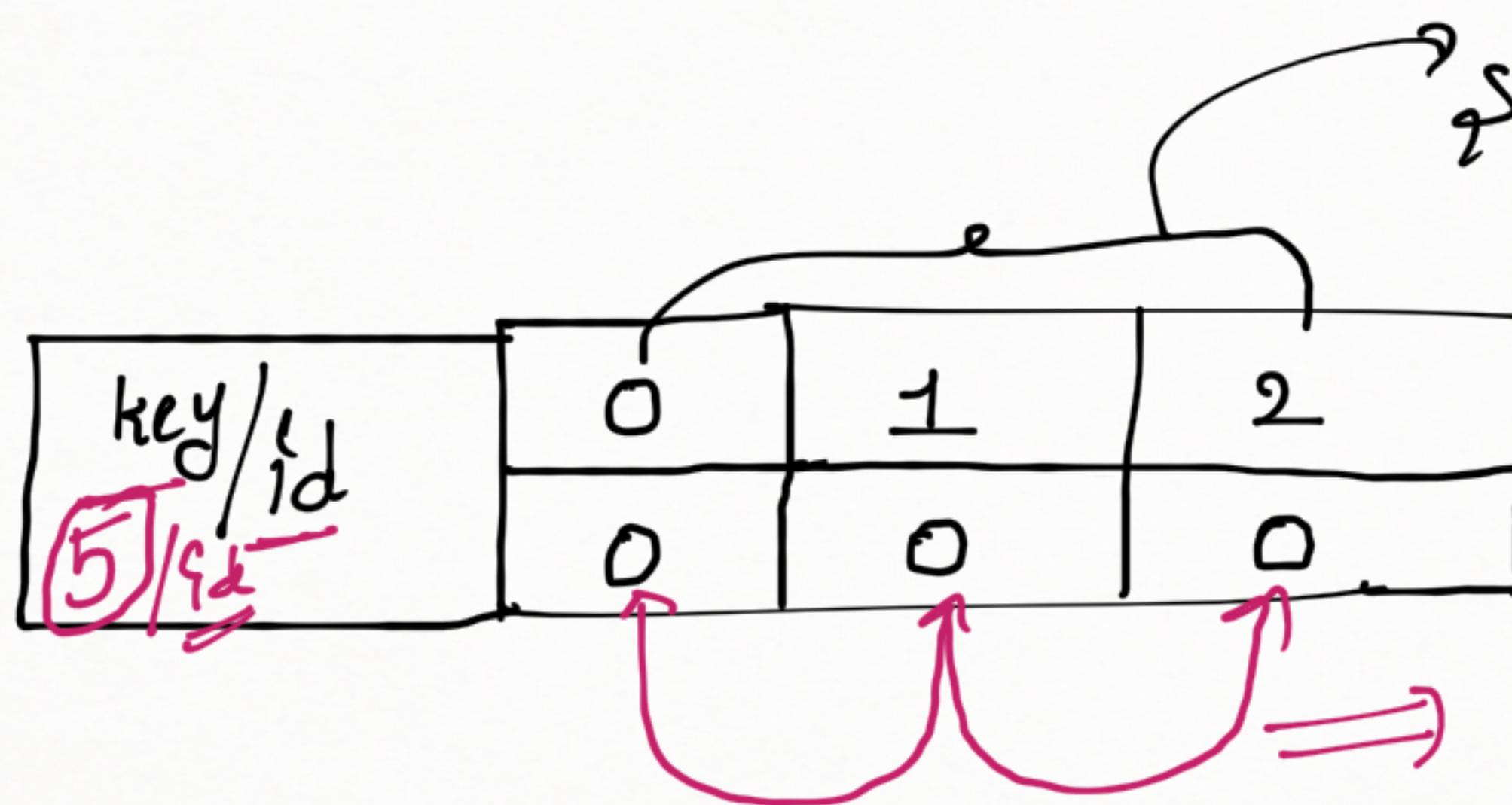


Date: 23-11-21

Semaphores

- Semaphore is one of the IPC but it is not for process communication. It is for process synchronization.
- Semaphore is a special object known by the kernel useful for process synchronization.
- Semaphore works based on Test (wait) and set (signal) mechanism (wait & signaling)

- By using command `IpcS -s` will display list of Semaphores.
- Function acquired for Semaphore operation are
 - (i) Semget() ⇒ useful for Semaphore creation
 - (ii) Semctl() ⇒ useful for Semaphore control operations
 - (iii) Semop() ⇒ useful for operations on semaphores



3 semaphores

0	1	2
---	---	---

all semaphores initial value
is zero

→ **semop** (int Semid, struct Sembuf *Sops, unsigned nops)

↓ ↓ ↓

semaphore identifier structure pointer no. of semaphores
Used

P1

int fd; struct sembuf v;

fd = Semget(5, 3, IPC_CREAT | 0644);

①
②
③

if (fd < 0)
 { perror("Semget");
 3 return;

v.sem_num = 0; // semaphore number zero

v.sem_op = 0; // Zero (or) +ve (or) -ve

(wait for
zero)

v.sem_flg = 0;

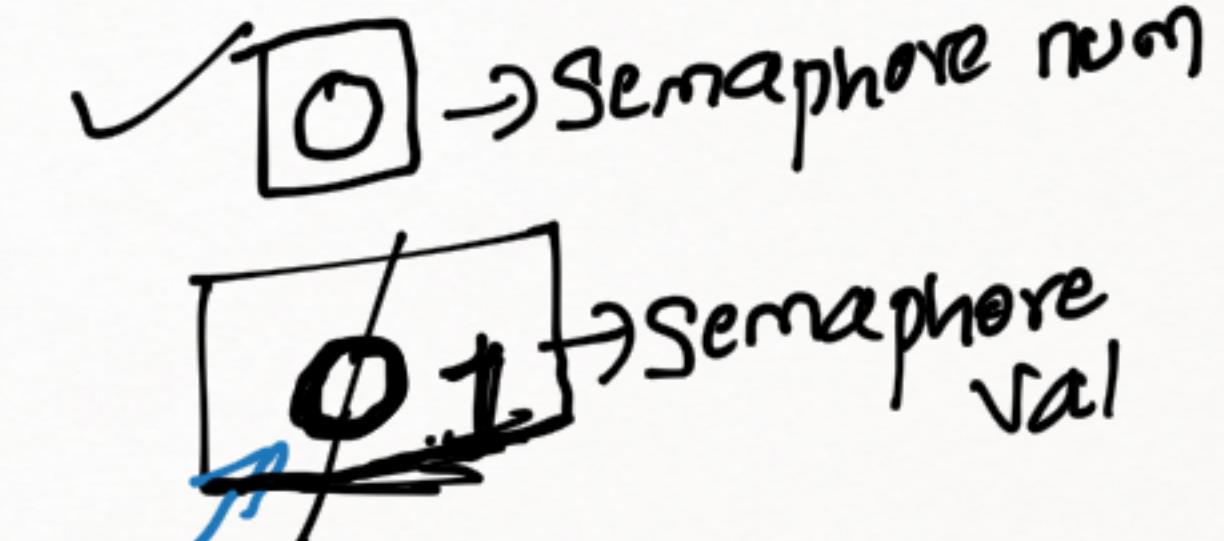
printf("Before semop--ln");

semop(fd, &v, 1);

✓ [] → (critical section)



→ Resource



(if sem-op value zero (wait for zero)
and semaphore value also zero
then process allowed)
(if sem-op value not zero then process blocked)

but semaphore
value not zero then process blocked

