

# Programación de software de sistemas

## Ayudantía 7

Profesor: Rodrigo Verschae

Ayudante: Nicolás Araya



# Semáforos

```
typedef struct {  
    int value;  
    pthread_mutex_t m;  
    pthread_cond_t c;  
} semaphore;
```





# Problema 1

Implemente las funciones

```
void semaphore_init(semaphore *s, int value),
```

```
void semaphore_destroy(semaphore *s),
```

```
void semaphore_wait(semaphore *s),
```

```
void semaphore_post(semaphore *s)
```

Aplicando la struct anterior.

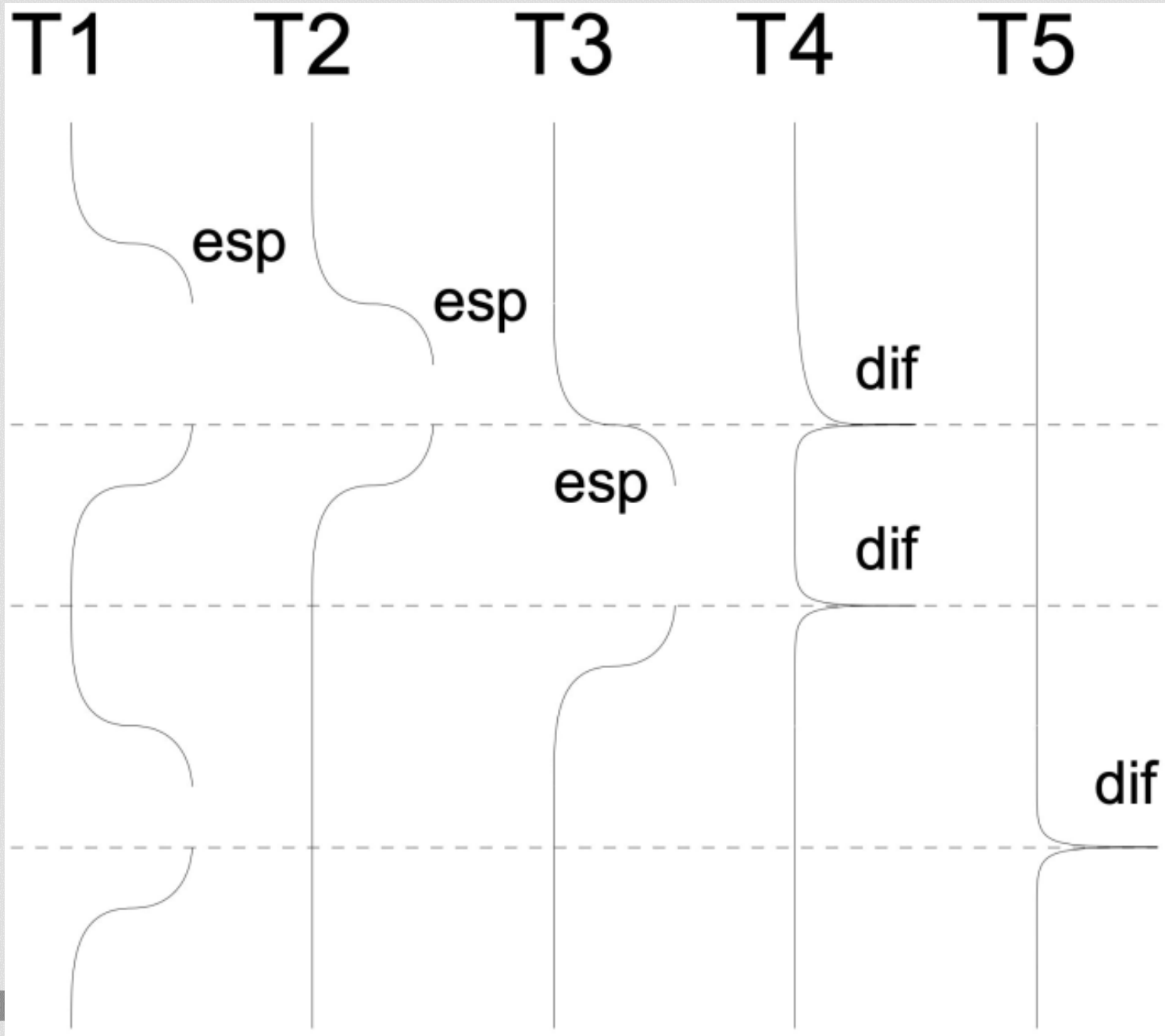




# Esperar y Difundir

- Existen 2 procedimientos:
  - Info\* esperar()
  - void difundir(Info\* info)
- Cuando un thread T1 invoca **esperar** se queda bloqueado hasta que otro thread T2 invoque **difundir**. Luego, T1 retorna la información dada por T2 a **difundir**.
- Si la invocación de **esperar** ocurre simultáneamente con **difundir** (traslape en las invocaciones), **esperar** puede retornar de inmediato o bloquearse hasta una nueva invocación de **difundir**.
- Si ocurren 2 invocaciones simultáneas de **difundir**, **esperar** puede retornar la información de cualquiera de esas 2 invocaciones a **difundir**.







# Solución propuesta

```
1
2
3
4
5
6
7
8
9
10 int cont = 0;
11 Info *info = NULL;
12 void difundir(Info *pinfo) {
13     cont++;
14     info = pinfo;
15 }
16
17 Info *esperar() {
18     int cont_actual = cont;
19     while (cont_actual == cont)
20         ;
21     return info;
22 }
23
24
25
26
27
28
29
30
31
32
33
```



# Problema 2

Haga un diagrama de threads que muestre que la solución propuesta es inconsistente y si es así proponga una mejor solución inspirada en la solución propuesta.

```
int cont = 0;
Info *info = NULL;
void difundir(Info *pinfo) {
    cont++;
    info = pinfo;
}
Info *esperar() {
    int cont_actual = cont;
    while (cont_actual == cont)
        ;
    return info;
}
```





# Problema 3

Escriba una solución consistente y eficiente de esperar y difundir usando como herramienta de sincronización los semáforos implementados en el problema 1.

Considere que estos semáforos garantizan que los semaphore\_wait serán atendidos en orden FIFO.





# Programación de software de sistemas

## Ayudantía 7

Profesor: Rodrigo Verschae

Ayudante: Nicolás Araya

