



Sistemas Operativos

Escuela de Ingeniería Civil Informática

- Administración de Procesos
- Sincronización con Mensajes



UNIVERSIDAD DEL BÍO-BÍO



MENSAJES

- El problema de los Semáforos es que se diseñaron para resolver el problema de la exclusión mutua en una o varias CPUs con memoria compartida.
- Si necesitamos trabajar sobre un sistema distribuido en la cual cada CPU utiliza su propia memoria, entonces estos mecanismos ya no se pueden aplicar.
- Para permitir el intercambio de información entre diferentes computadores de una red, se ideó un **sistema de mensajes**.
- Típicamente los mensajes se implementan como una librería, que es parte de un lenguaje de programación o incluso del sistema operativo.





MENSAJES EN N-SYSTEM

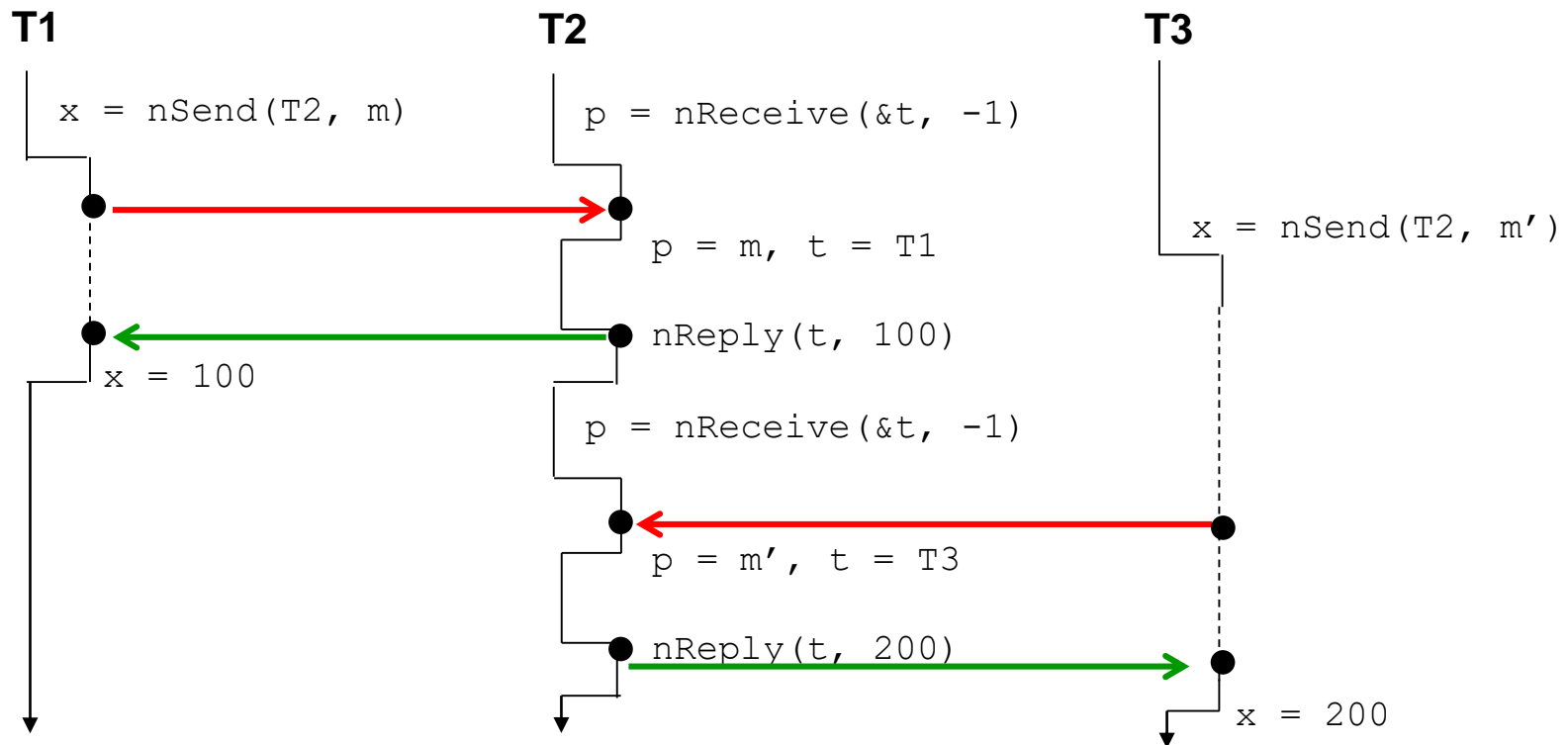
- Esta librería se compone de 3 funciones:
 - ***int nSend(nTask task, void *msg);***
 - Envía el mensaje *msg* a la tarea *task*.
 - Un mensaje consiste en un puntero a un área de datos de cualquier tamaño.
 - *nSend* se queda bloqueado hasta que *task* reciba el mensaje (invocando *nReceive*) y además lo responda (invocando *nReply*).
 - *nSend* retorna un entero especificado en *nReply*.
 - ***void *nReceive(Ntask *ptask, int delay);***
 - Espera y recibe un mensaje enviado por cualquier otra tarea
 - Si *delay* $\geq 0 \rightarrow$ espera *delay* microsegundos
 - Si *delay* = -1 \rightarrow espera ilimitadamente hasta que llegue un mensaje
 - *nReceive* retorna el puntero *msg* especificado por el emisor y deja en **ptask* la identificación del emisor. En caso contrario, *nReceive* se desbloquea a los *delay* microsegundos y retorna NULL.
 - *nReceive* recibe mensajes en orden FIFO de envío.
 - El mensaje retornado por *nReceive* es un puntero a un área que ha sido posiblemente creada en la pila del emisor. Dado que el emisor continúa bloqueado hasta que se invoque a *nReply*, el receptor puede acceder libremente esta área sin el peligro de que sea destruída por el emisor.



MENSAJES EN N-SYSTEM

- ***void nReply(nTask task, int rc);***
 - Responde un mensaje enviado por *task*.
 - Desde ese instante, el receptor no puede acceder la información contenida en el mensaje que había sido enviado, ya que el emisor podría destruirlo.
 - El valor *rc* es el código de retorno para el emisor.

DIAGRAMA DE TAREAS DE SU FUNCIONAMIENTO



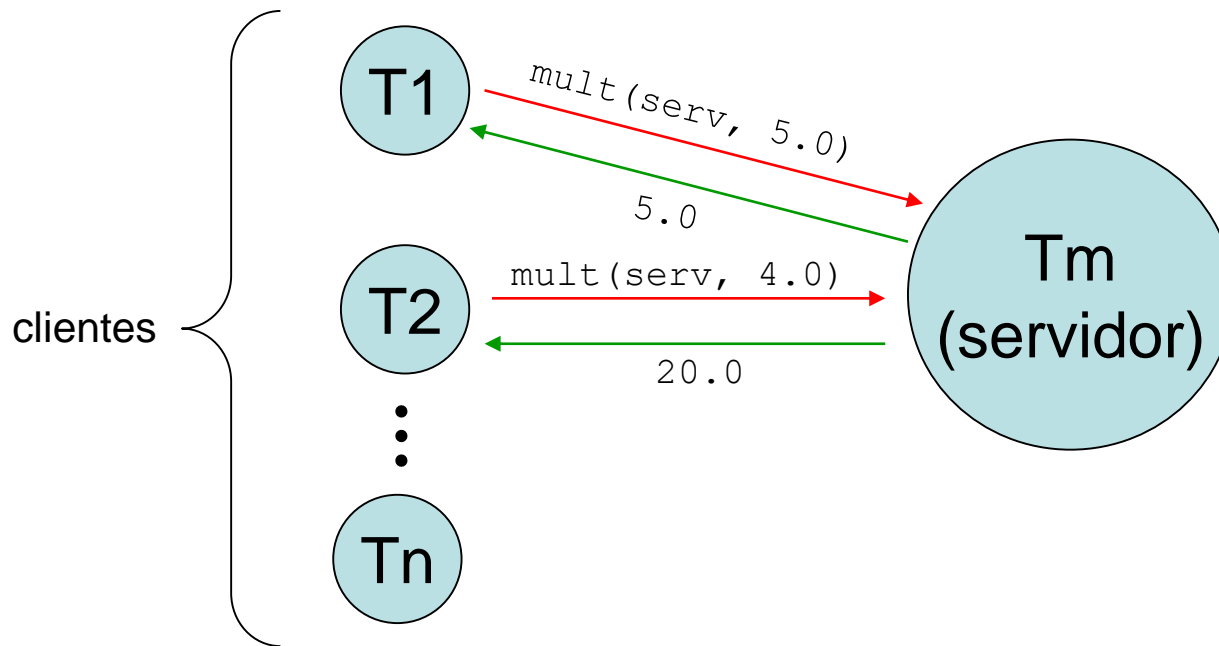


EJEMPLO: MENSAJES EN N-SYSTEM

MULTIPLICADOR CONCURRENTES (MT - Safe)

Varios threads pueden invocar concurrentemente el multiplicador teniendo un thread exclusivo que realiza las multiplicaciones.

Diseño de la solución



Ejemplo de uso

```
Mult m= makeMult();
p = mult(m, 5.0);
...
destroyMult(m);
```



EJEMPLO: MENSAJES EN N-SYSTEM

Implementación de la solución

```
#include <nSystem.h>
```

```
#define MULT 1
```

```
#define DEST 2
```

```
typedef struct {  
    nTask serv;  
} *Mult;
```

```
typedef struct {  
    int code;  
    double f, p;  
} Req;
```

```
Mult makeMult() {  
    Mult m = (Mult) nMalloc(sizeof *m);  
    m->serv = nEmitTask(multProc);  
    return m;  
}
```

```
double mult(Mult m, double f) {  
    Req rq;  
    rq.code = MULT;  
    rq.f= f;  
    nSend(m->serv, &rq);  
    return(rq.p);  
}
```

```
void destroyMult(Mult m) {  
    Req rq;  
    rq.code = DEST;  
    nSend(m->serv, &rq);  
    nWaitTask(m->serv);  
    nFree(m);  
}
```



EJEMPLO: MENSAJES EN N-SYSTEM

```
void multProc() {  
    double p = 1.0;  
    for(;;) {  
        Req *prq;  
        nTask t;  
        prq = (Req *) nReceive(&t, -1);  
        if (prq->code == MULT) {  
            p = p * prq->f;  
            prq->p = p;  
            nReply(t, 0);  
        }  
        else { /*DEST*/  
            nReply(t, 0);  
            nExitTask(0);  
        }  
    }  
}
```




EJERCICIO : EL PUENTE

Se tiene un puente de una sola vía por donde pasan autos en ambos sentidos. Cuando un auto llega por el norte invoca el procedimiento *EntraNorte()* y al salir invoca *SaleSur()*. De la misma forma los autos que viajan de sur a norte invocan *EntraSur()* y *SaleNorte()* respectivamente para entrar y salir. Los automovilistas se quejan de lo incómodo que resulta cuando se encuentran en medio del puente con otro auto que viene en sentido contrario. Por lo tanto, se le ha encomendado a Ud. sincronizar la entrada al puente de modo de evitar que dos automóviles se encuentren en el puente viajando en sentidos opuestos. Ud. debe permitir que ingresen al puente varios automóviles viajando en el mismo sentido.

Importante:

- No se preocupe por situaciones de hambruna.
- Para lograr la sincronización utilice mensajes y tareas de nSystem.

