Auxiliar ?+2 - CC3301

Alonso González Ulloa

July 2, 2010

Problema 1 (Los Juntapuchos)

En alguna acera existe un grupo de mendigos conocidos como "Los Juntapuchos". El apodo no es casual, pues de hecho su principal preocupación es juntar una determinada cantidad de colillas de cigarrillos que bota la gente, para sacarle el tabaco y fumarse un buen cigarro.

La situación anterior puede ser modelada usando threads que acceden a recursos compartidos, de este modo cada Juntapuchos es un thread y los recursos compartidos son las colillas. Suponga que existen `NJP` Junta Puchos y con `NCOLILLAS` colillas hacen un pucho. Suponga que inicialmente existen `INITIAL_COLILLAS` colillas y que hay un tercer actor que tiene muchos cigarrillos, y solo se dedica a fumarlos botando una colilla por cada cigarrillo fumado.

Para modelar cada Juntapuchos use la siguiente estructura

```
typedef struct {
     pthread_t pid;
     char *name;
     colillas_t *colillas;
} junta_puchos;
```

Donde colillas es un puntero a una estructura que representa a las colillas que se encuentran en la acera, definida como sigue

```
typedef struct {
     pthread_mutex_t s_colillas;
     pthread_cond_t condition;
     int ncolillas;
} colillas_t;
```

Considere al actor con muchos cigarrillos como un Juntapuchos más, o si lo desea

```
typedef junta_puchos fumador
```

Se le pide programar las siguientes funciones

```
/* Inicializa un junta_puchos y lo hace partir */
junta_puchos *make_junta_puchos(char *, colillas_t *);
```

```
/* Permite a un junta_puchos obtener la cantidad
 * necesaria de colillas para hacer un pucho.
 * Como los junta_puchos solo hacen puchos, esta
 * función debe correr eternamente.
 */
void get_colillas(junta_puchos *);

/* Una vez que tiene las colillas, el junta_puchos
 * hace el pucho y se lo fuma relajadamente tomandose
 * cierto tiempo.
 */
void make_pucho(junta_puchos *);
```

Adicionalmente programe las siguientes funciones

```
/* Inicializa a un fumador y lo hace partir */
fumador *make_fumador(char *, colillas_t *);

/* Hace que el fumador fume y fume sin parar
 * botando una colilla por cada cigarro fumado
 */
void fumar(fumador *);
```

Programe las funciones de tal forma que no hayan *"data races"* ni *"deadlocks"*

Problema 2 (Servidor de Chat)

Programe un servidor de Chat de modo tal que la conversación de dos clientes se ve así:

- Conexión del primer cliente

```
user1@pc1:~$ telnet serverchat.cl 1818
Trying 192.80.24.6...
Connected to serverchat.cl.
Escape character is '^]'.
Escriba su nick user1
```

- Conexión del segundo cliente

```
user2@pc2:~$ telnet serverchat.cl 1818
Trying 192.80.24.6...
Connected to serverchat.cl.
Escape character is '^]'.
Escriba su nick user2
```

- user1 escribe “Hola user2”, enotonces user 2 debe ver lo siguiente

```
user1:Hola user2
```

- Del mismo modo, si user2 escribe “Hola user1” entonces user1 ve lo siguiente

```
user2:Hola user1
```

Programe el servidor usando `select` y `jsocket`. Considere que se pueden conectar hasta simultaneamente hasta `NCLIENTS` clientes.