

Utilisation des B.A.L (File de messages) Unix

1. Introduction

- L'utilisation des boîtes aux lettres (B.A.L) pour la synchronisation avec échange de messages, suppose défini un ensemble de constantes et de types d'objet (`key_t`, `msqid_ds`, ...). Ces définitions sont obtenues en ajoutant les directives d'inclusion suivantes en tête des programmes utilisant les B.A.L:

```
#include <sys/types.h>
```

```
#include <sys/ipc.h>
```

```
#include <sys/msg.h>
```

Utilisation des B.A.L Unix

- Chaque boîte aux lettres est décrite par son descripteur de type `msgid_ds`. Quatres primitives *msgget* , *msgctl*, *msgsnd* et *msgrcv* contrôlent la communication par boîtes aux lettres.
- Rem: Des processus n'ayant aucun lien de parenté peuvent communiquer par BAL, ce qui n'est pas le cas des tubes anonymes (en mémoire).

Utilisation des B.A.L Unix

2. Ouverture d'une boîte aux lettres

- La primitive *msgget* consiste à attribuer un numéro interne à une boîte aux lettres désignée par un nom externe.

int msgget(Nom_Ext, Indicateur)

key_t Nom_Ext ;

int Indicateur ;

- Selon la valeur du paramètre *Indicateur*, la primitive *msgget* réalise la création ou l'ouverture de la boîte aux lettres.

Utilisation des B.A.L Unix

a) La création

- Elle est effectuée si le nom externe a la valeur particulière `IPC_PRIVATE` (`((key_t) 0)`), ou si le bit `IPC_CREAT` de `Indicateur` est à 1. Dans ce dernier cas, le bit `IPC_EXCL` joue le même rôle que pour les sémaphores. S'il est positionné, la création échoue si le nom externe est déjà utilisé, s'il ne l'est pas l'appelant risque d'obtenir le numéro interne d'une boîte aux lettres créée par une autre application.
- Les droits d'accès à la boîte aux lettres créée sont indiqués dans les 9 bits de poids faible de *Indicateur*.

Utilisation des B.A.L Unix

b) L'ouverture

Lors d'un appel de la primitive *msgget*, pour obtenir le numéro interne, la valeur du bit `IPC_CREAT` importe peu pourvu que `IPC_EXCL` et `IPC_CREAT` ne soient pas tous les deux positionnés.

Utilisation des B.A.L Unix

3. Contrôle des attributs d'une boîte aux lettres

- La primitive *msgctl* permet des opérations de contrôle sur une boîte aux lettres, telles que:
 - Le changement des droits d'accès du propriétaire ou du groupe.
 - La consultation des caractéristiques de la boîte aux lettres.
 - La destruction de la boîte aux lettres.

Utilisation des B.A.L Unix

- La valeur retournée par la primitive `msgctl` est 0 en cas de succès et -1 en cas d'échec.

***int** msgctl (Num_Int, Controle, Tampon)*

***int** Num_int, Controle;*

***struct** msqid_ds *Tampon; //dans msg.h*

Utilisation des B.A.L Unix

- Les valeurs possibles de Controle sont:
 - IPC_STAT, copie le descriptif de la boîte aux lettres dans Tampon.
 - IPC_SET, modifie certaines caractéristiques de la boîte aux lettres à partir des valeurs contenues dans Tampon.
 - IPC_RMID, détruit la boîte aux lettres.

Utilisation des B.A.L Unix

- A la suite de la destruction d'une BAL, tout processus mis en attente à cause de l'état de la boîte aux lettres qui ne permet pas d'exécuter l'opération qu'il demande, est alors réveillé et averti de cet événement (Il reçoit -1 et une erreur EIDRM (36) dans *errno*).

Utilisation des B.A.L Unix

4. Les échanges de message (Dépôt/Retrait)

- Dans la description des primitives permettant l'échange de messages, le type `msgbuf` est utilisé :

```
struct msgbuf
```

```
{ long mtype; /* type du message */  
  char mtexte[ ]; /* texte du message */  
}
```

Utilisation des B.A.L Unix

a) Le dépôt de message

- Un producteur dépose un message dans une boîte aux lettres en utilisant la primitive *msgsnd*.

int msgsnd (Num_Int, Mes, Lg_Mes, Indic)

int Num_Int, Lg_Mes, Indic;

struct msgbuf *Mes;

Utilisation des B.A.L Unix

- Le dépôt peut-être momentanément bloqué si la boîte aux lettres est pleine ou ne contient pas suffisamment de place libre. Si cette situation se produit, le bit `IPC_NOWAIT` de *Indic* précise l'action à entreprendre.
 - `IPC_NOWAIT=0` (valeur par défaut) : bloquer l'appelant jusqu'à ce qu'un retrait soit effectué
 - `IPC_NOWAIT=1` : laisser l'appelant se poursuivre en l'avertissant que son message n'a pas été déposé (retour de -1 et `EAGAIN(11)` dans *errno*).

Utilisation des B.A.L Unix

b) Le retrait de message

- Un consommateur retire un message d'une boîte aux lettres en utilisant la primitive msgrcv.

```
int msgrcv (Num_Int, Mes, Lg_Mes,  
Type_Mes, Indic)
```

```
int Num_Int, Lg_Mes, Indic;
```

```
struct msgbuf *Mes;
```

```
long Type_Mes;
```

Utilisation des B.A.L Unix

- Le type de message demandé par l'appelant est spécifié selon les conventions suivantes:
 - $Type_Mes = 0$, Premier message quelque soit son type (FIFO).
 - $Type_Mes > 0$, Premier message ayant pour type $Type_Mes$.
 - $Type_Mes < 0$, Premier message dont le type est choisi par ordre de préférence parmi $1, 2, \dots | Type_Mes |$.

Utilisation des B.A.L Unix

- La valeur retournée par *msgrcv* est la longueur du message délivré à l'appelant.
- Si la longueur du message dans *msgrcv* est inférieure à la longueur du message dans *msgsnd*, le bit MSG_NOERROR du paramètre *Indic* dicte la conduite à tenir.
 - MSG_NOERROR = 1, tronquer le message en gardant que les Lg_Mes premiers caractères.
 - MSG_NOERROR = 0 (valeur par défaut), signaler à l'appelant l'impossibilité du retrait (-1 comme valeur retournée et E2BIG(7) dans *errno*)
- Rem: un message lu en partie est considéré comme étant lu en entier.

Utilisation des B.A.L Unix

- Le retrait d'un message peut-être momentanément bloqué si la boîte aux lettres ne contient aucun message d'un des types demandés. Si c'est le cas, le bit `IPC_NOWAIT` indique l'action à entreprendre.
 - `IPC_NOWAIT = 0` (valeur par défaut): Bloquer l'appelant jusqu'à ce qu'un message d'un des types demandés soit déposé.
 - `IPC_NOWAIT = 1` : Laisser l'appelant se poursuivre en l'avertissant de l'échec de sa demande (-1 comme valeur retournée et `EAGAIN(11)` dans `errno`).