Département Informatique INSA de Lyon

Conception Multitâches

(LACATRE / Vx Works)

Jean-Philippe Babau

Département Informatique, laboratoire CITI
INSA Lyon

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

jean-philippe.babau@insa-lyon.fr

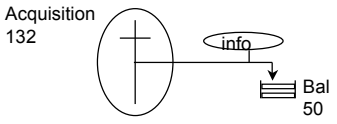
LACATRE

- Abréviation de
  - Langage d'Aide à la Conception d'Applications Temps Réel
- Langage graphique
  - version textuelle
- Langage de programmation des ETR
- Facilite la communication entre concepteurs
- *Génération automatique de code*
- Vérification de propriétés
  - Création / destruction

jean-philippe.babau@insa-lyon.fr

LACATRE : principe

Forme graphique



Forme textuelle

```
TASK(Acquisition,132)
      FOREVER
      send_to_mbx(Bal,info,,);
END_TASK
```

```
...
int Acquisition(void)
{ /* local varaibles of the task < Acquisition> */
      /* body of the task <Acquisition> */
      for (;;)
      {
            msgQSend(Bal,(char*)&info,sizeof(info)

,WAIT_FOREVER,MSG_PRI_NORMAL);
      }
}
```

Code C/VxWorks

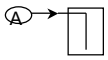
jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

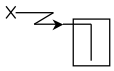
LACATRE : les objets

- Les objets programmables

la tâche



l'alarme



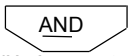
la routine d'interruption

- Les objets configurables

le sémaphore

la boîte aux lettres

la ressource



l'événement

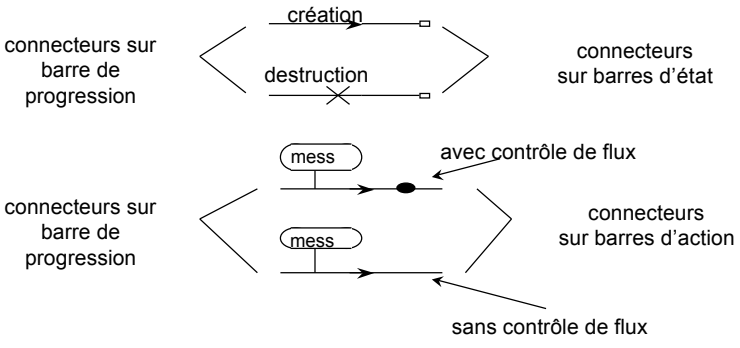
message,
paramètres

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

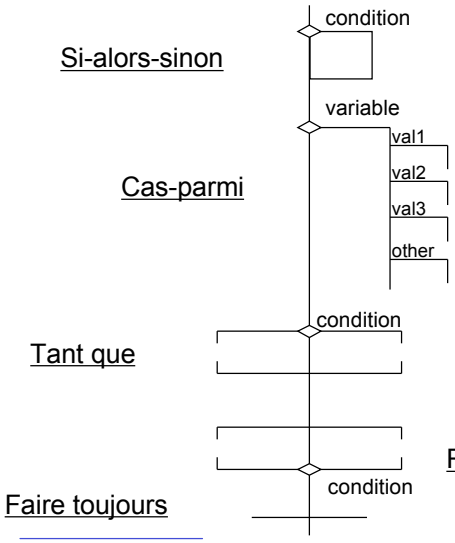
LACATRE : les actions

- Symbole graphique
  - les connecteurs
  - lignes de liaison
  - décor



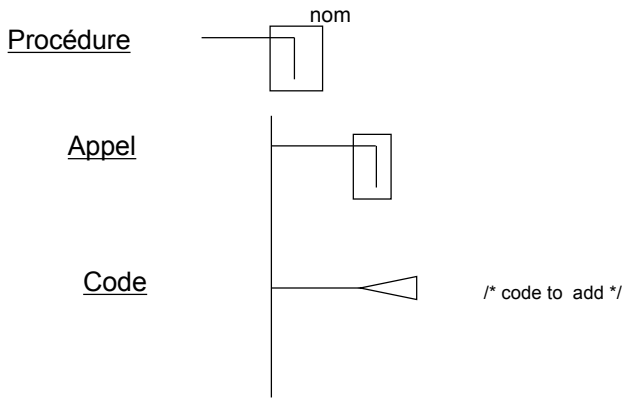
jean-philippe.babau@insa-lyon.fr

LACATRE : formes algorithmiques



jean-philippe.babau@insa-lyon.fr

LACATRE : formes algorithmiques

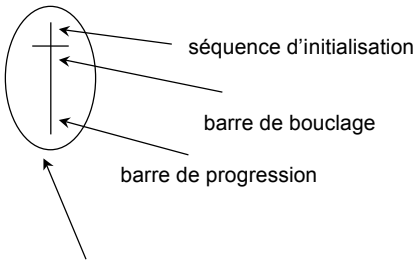


- Les aspect transformationnels sont traités en C
- Les structures sont liés aux autres objets temps réel

jean-philippe.babau@insa-lyon.fr

LACATRE : la tâche

nomTache
priorite ← paramètres textuels



C/VxWorks

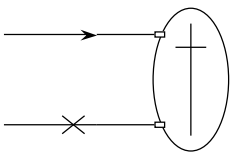
```
LOCAL int  taskId ;

int nomTache(void)
{
/* declaration variables locales*/
/* instructions d'initialisation */
for ( ; ; ) /* forever */
{
      /* instructions */
      /* attente */
      /*instructions */
}
return 0}
```

jean-philippe.babau@insa-lyon.fr

LACATRE : la tâche

Création

Destruction



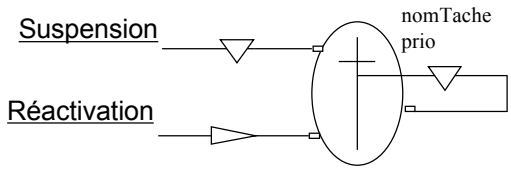
C/VxWorks

```
LOCAL int  taskId ;
taskId =taskSpawn("AFF",prio,0,10000, NomTache,0,0,0,0,0,0,0,0,0,0);
status = taskDelete(taskId );
exit() ; /* autodestruction */
```

- Destruction lorsque toutes les ressources sont libérées
- Protection contre la destruction ! attention !
  - taskSafe ();
  - taskUnSafe();

jean-philippe.babau@insa-lyon.fr

LACATRE : la tâche

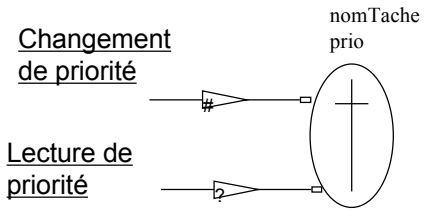


C/VxWorks

```
taskSuspend(taskId) ;
taskSuspend(0); /* autosuspension */
taskResume(taskId) ;
```

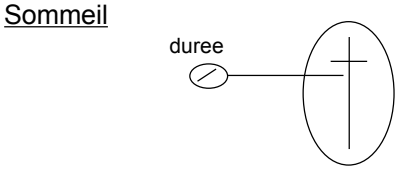
jean-philippe.babau@insa-lyon.fr

LACATRE : la tâche



C/VxWorks

```
status = taskPrioritySet(taskId,,priority) ;
status = taskPriorityGet(taskId, &priority) ;
```



C/VxWorks

```
taskDelay(duree);
/* en ticks (60 ticks par secondes par defaut)*/
```

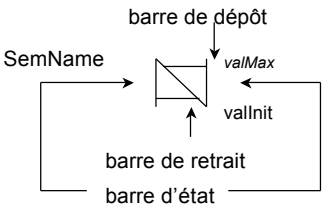
jean-philippe.babau@insa-lyon.fr

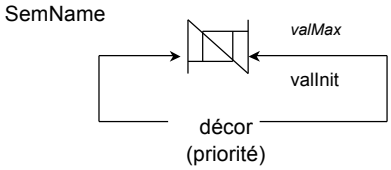
Autres primitives

- Pas de représentation graphique
- nom = taskName(taskId)
  - renvoie le nom de la tâche (char*)
- takId = taskNameToId(" AFF")
  - renvoie l'id d'une tâche pour un nom donné
- takId = taskIdSelf()
  - renvoie l'id de la tâche appelante
- Existe = taskIdVerify(taskId)
  - vérifie l'existence d'une tâche
- taskIdListGet(); taskisReady(taskId) ; …

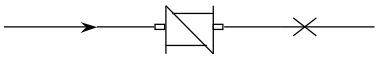
jean-philippe.babau@insa-lyon.fr

LACATRE : le sémaphore





Création



Destruction

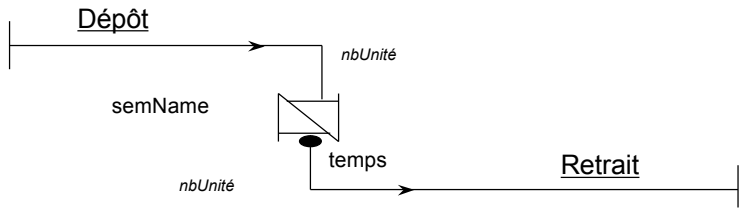
```
semId = semBCreate(SEM_Q_FIFO | SEM_Q_PRIORITY,
                   SEM_FULL | SEM_EMPTY) ;
semId = semCCreate(SEM_Q_FIFO | SEM_Q_PRIORITY, initCount) ;
status = semDelete(semId) ;
```

C/VxWorks

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

LACATRE : le sémaphore



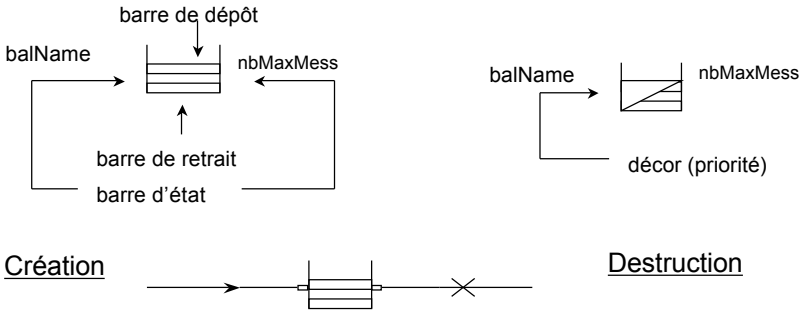
C/VxWorks

```
status = semGive(semId) ;
status = semFlush(semId) ; /* déblocage de toutes les tâches en attente */

status = semTake(semId, temps | WAIT_FOREVER | NO_WAIT) ;
```

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

LACATRE : la boîte aux lettres

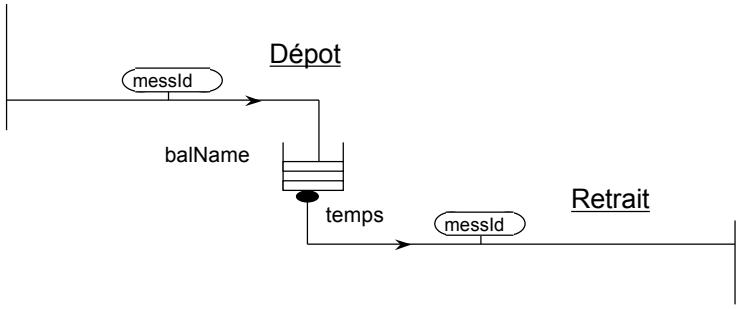



```
msgQId = msgQCreate(nbMaxMess, longueurMax,
                    SEM_Q_FIFO | SEM_Q_PRIORITY) ;
status = msgQDelete(msgQId) ;
```

C/VxWorks

jean-philippe.babau@insa-lyon.fr

LACATRE : la boîte aux lettres

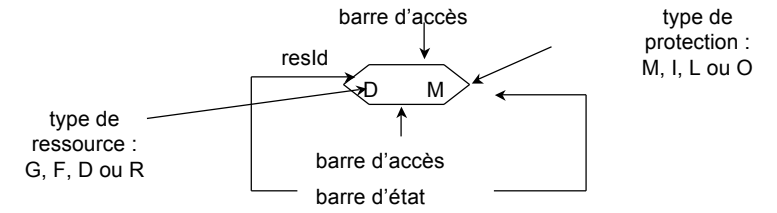



```
status = msgQSend(msgQId,buffer,nOctets,temps | NO_WAIT
| WAIT_FOREVER, MSG_PRI_NORMAL | MSG_PRI_URGENT ) ;

val = msgQReceive(msgQId,buffer,nOctets,temps |
                        WAIT_FOREVER | NO_WAIT ) ;
/* renvoie le nombre d'octets copiés */
```

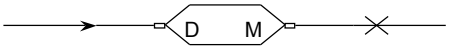
C/VxWorks

jean-philippe.babau@insa-lyon.fr

LACATRE : la ressource



Création



Destruction

```
semId = semMCreate(SEM_Q_FIFO | SEM_Q_PRIORITY |
            SEM_DELETE_SAFE | SEM_INVERSION_SAFE) ;
/* etat SEM_FULL */
status = semDelete(semId) ;
```

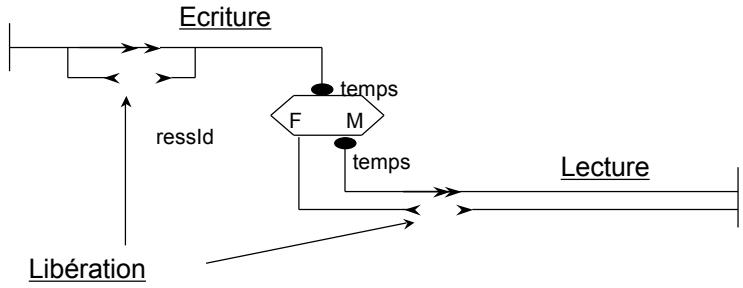
C/VxWorks

si protection par sémaphore

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

LACATRE : la ressource




```
status = semTake(semId, temps |
WAIT_FOREVER | NO_WAIT) ;

status = semGive(semId) ; /* semFlush interdit */
```

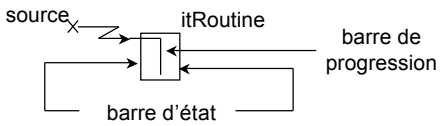
C/VxWorks

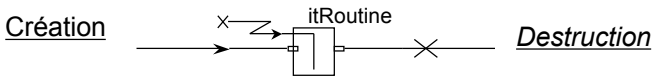
si protection par sémaphore

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

LACATRE : l 'interruption





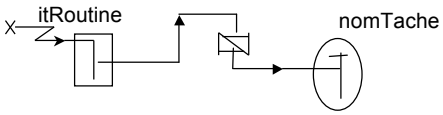

```
intConnect(adresseVecteur,itRoutine,parametre) ;

itRoutine(int val)
{ … }
```

C/VxWorks

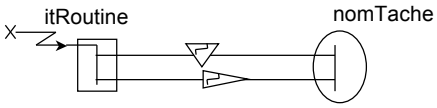
jean-philippe.babau@insa-lyon.fr

LACATRE : l 'interruption

Déclenchement



Masquage / démasquage



C/VxWorks

```
lock = intLock() ;
intUnLock (lock) ;
```

C/VxWorks

```
status = intDisable (level) ;
status = intEnable ( level) ;
```

jean-philippe.babau@insa-lyon.fr

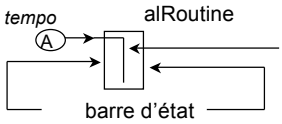
Interruptions sous VxWorks

- Primitives autorisées
  - semGive
  - msgQSend
  - write (pipe)
  - taskSuspend
  - wdStart, wdCancel
- **Primitives interdites : primitives bloquantes**
  - msgQReceive, semTake
  - semGive sur un sémaphore d'exclusion mutuelle, semFlush
  - taskLock
  - printf (appel d'une primitive bloquante)
- Interruptions logicielles
  - getchar () ;
  - control C

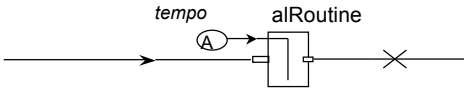
jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

LACATRE (*extensions*) : l'alarme



Création



Destruction

```
LOCAL WDOG_ID   wdId ;
wdId = wdCreate();

status = wdDelete(wdId ) ;
```

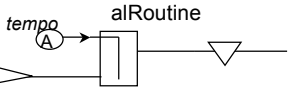
C/VxWorks

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

LACATRE (*extensions*) : l'alarme

Activation

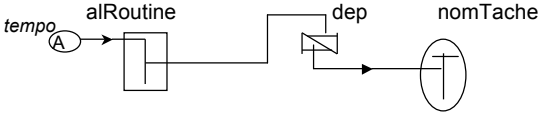


Arrêt

```
status = wdStart(wdId,tempo,alRoutine,param);

status = wdCancel(wdId);
```

C/VxWorks

Déclenchement



jean-philippe.babau@insa-lyon.fr

Et les autres RTOS …

- Chaque RTOS défini des objets et des services spécifiques
  - Cf. cours OS pour l'embarqué
- Manipulation de concepts de haut niveau
  - Par exemple la ressource LACATRE
  - À implémenter avec le RTOS utilisé
- Manipulation de concepts de bas niveau
  - Concept spécifique à un cible
  - Comportement spécifique à une cible
    - Sémaphore à compte de VxWorks : pas de prise/libération de n jetons
- Portage des architectures multitâches limité
  - Concepts
  - Formalisme graphique

jean-philippe.babau@insa-lyon.fr

Conception

- Principes
- Aspect réactif et concurrents
  - Tâches
  - Communication avec l'environnement externe
  - Échange d'information
- Prédictibilité
  - Statique / dynamique
- Vérification, mise au point
- Distribution

jean-philippe.babau@insa-lyon.fr

Principes généraux

- Spécification
  - Un événement externe ➔ séquence d'actions
  - Contraintes temps réel (échéances)
  - Politiques de réaction aux fautes temporelles
  - Reconfiguration
- Mise en place de tâches
  - Réalisation des actions
  - Lien avec l'environnement
  - Respects des contraintes temps réel
    - Priorité
  - Phases de l'application
    - Création / destruction / suspension / réactivation
    - Moteur / exception
  - Réutilisation
    - Processus
    - Pilotes, applications, …

jean-philippe.babau@insa-lyon.fr

Type de tâches

- Initiale
  - activée au lancement de l'application
- Ordinaire
  - liée à un événement externe ou interne
  - tâches matérielles
    - liée à une alarme périodique
    - liée à une interruption
  - boucle infinie avec attente
    - sémaphore (binaire ou à compte)
    - boîte aux lettres
    - Mécanisme spécifique de synchronisation

- tâches logicielles
  - activée par une autre tâche

- Tâche de fond
  - toujours active
  - exécutée lorsque toutes les autres tâches sont terminées

jean-philippe.babau@insa-lyon.fr

Mise en place de l'architecture logicielle

- Déclenchement des tâches
- Exécution des actions
  - Prise en compte des événements
  - Concurrence et coopération
  - Contraintes de temps
  - Initialisation des données et services
- Échange de données
- Création / destruction des éléments

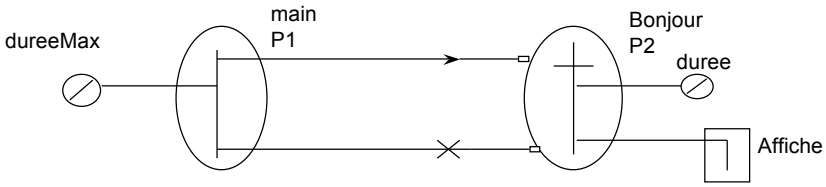
jean-philippe.babau@insa-lyon.fr

Déclenchement

- Traitement périodique
  - Traitement régulier
  - Traitement du signal
  - Scrutation d'un procédé ou de l'environnement
- Nombres d'activation
- Synchronisation des actions
  - Plusieurs tâches périodiques
- Traitement à date fixe
- Traitement des interruptions

jean-philippe.babau@insa-lyon.fr

Exemple 1 (LACATRE)




```
void Affiche( int valeur )
{           printf(" temps = %d\n ", valeur) ;
}
```

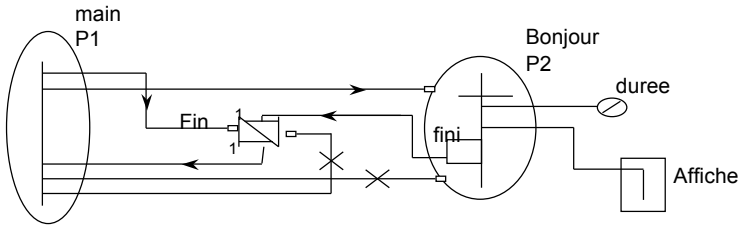
jean-philippe.babau@insa-lyon.fr

Exemple 1 (VxWorks)

```
#include <taskLib.h>        #include <time.h>

#define P2 150              #define dureeMax 150        #define duree 10

LOCAL int BonjourId ;
int Bonjour(void);

int main(void)
{
BonjourId = taskSpawn(" TASK1",P2,0,20000,Bonjour,0,0,0,0,0,0,0,0,0,0);
taskDelay(dureeMax);
taskDelete(BonjourId );
return (0); }

int Bonjour(void)
{    int temps = 0;
     for(;;)
     { taskDelay(duree);
       Affiche( temps ++);  }
return 0; }
```

jean-philippe.babau@insa-lyon.fr

Exemple 2 (LACATRE)



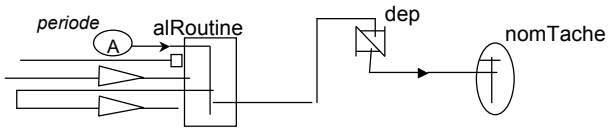
jean-philippe.babau@insa-lyon.fr

Exemple 2 (VxWorks)

```
int main()
{

FinId = SemBCreate(SEM_Q_FIFO, EMPTY) ;
BonjourId =
taskSpawn(" TASK1",P2,0,20000,Bonjour,0,0,0,0,0,0,0,0,0,0);
semTake(FinId, WAIT_FOREVER) ;
taskDelete (EcrireId ) ;
semDelete (FinId ) ;
return(0) ;
}

int Bonjour()
{
      int temps = 0 ;
      for ( ; ; )
      {       taskDelay(duree) ;
              Affiche( temps++ ) ;
/* fini */    if (temps == 5 )
                    semGive(FinId) ;
      }
return 0 ; }
```

jean-philippe.babau@insa-lyon.fr

Tâche périodique



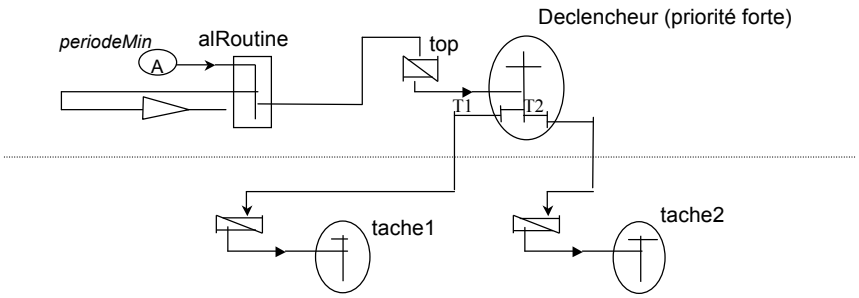

```
#define periode 100
…
wdId = wdCreate();
wdStart(wdId,0,alRoutine,0);
…
alRoutine ( int valeur)
{           wdStart(wdId,periode,alRoutine,0);
            semGive(dep) ;
}
```

C/VxWorks

jean-philippe.babau@insa-lyon.fr

Déclenchement synchronisé

Contôle (~RTOS)



Application

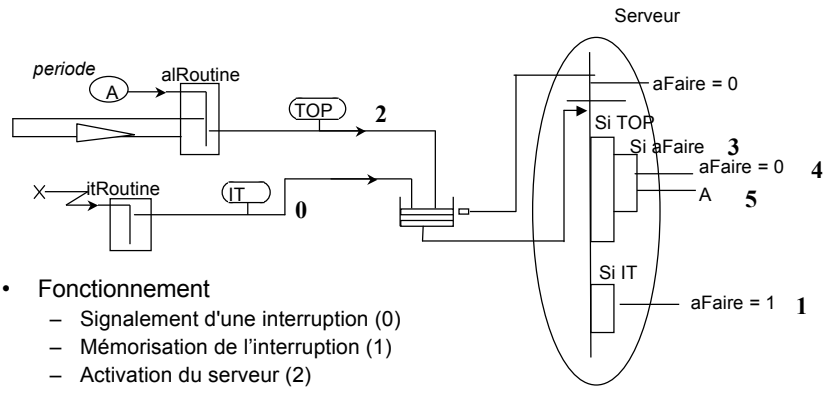
jean-philippe.babau@insa-lyon.fr

Traitement des interruptions

- Routine d'IT
  - pas de traitements
  - mémorisation des its
  - acquisition
  - datation
- Serveur d'IT
  - traitement lié à l'IT
  - immédiat / différé
- Communication routine/serveur
  - sémaphore booléen
  - sémaphore à compte
  - boite aux lettres
  - donnée sans protection
    - masquage des IT

jean-philippe.babau@insa-lyon.fr

Serveur à scrutation



- Fonctionnement
  - Signalement d'une interruption (0)
  - Mémorisation de l'interruption (1)
  - Activation du serveur (2)
  - Test sur interruption à traiter (3)
  - Acquittement logiciel (4)
  - Traitement (5)
- Variantes
  - Afaire ++, si aFaire A(n)

jean-philippe.babau@insa-lyon.fr

Scrutation / événementiel

- Événementiel
  - évaluation de dmin
    - dynamique du procédé
    - valeur imposée
  - Protection
    - masquage/démasquage
  - Pas de perte de temps
- Scrutation
  - L'application définit son rythme de travail
  - Prédictibilité
  - Obligatoire dans certains domaines (certification)
  - OSEK Time

jean-philippe.babau@insa-lyon.fr

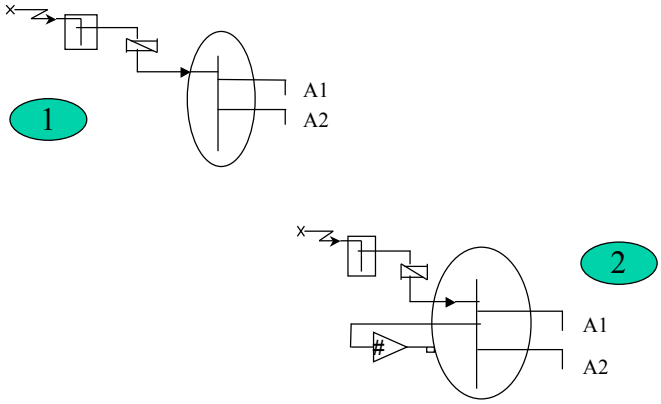
Mise en place des tâches pour l'exécution des actions

- Actions
  - À partir d'un événement externe (alarme ou interruption) : suivi du fil d'exécution
  - Une tâche = une séquence d'actions
- Gestion des contraintes de temps
  - Priorités des actions -> priorités des tâches
  - Priorité liée à l'urgence temporelle (cf. cours d'ordonnancement)
- Gestion des phases et des composants
  - Initialisation / urgence / …
  - Découpage en processus

jean-philippe.babau@insa-lyon.fr

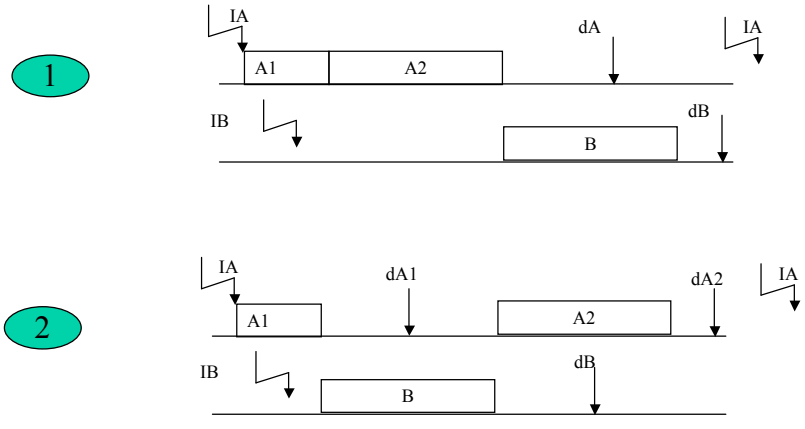
Tâches Vs actions (1)

- 2 actions en séquence A1 – A2



jean-philippe.babau@insa-lyon.fr

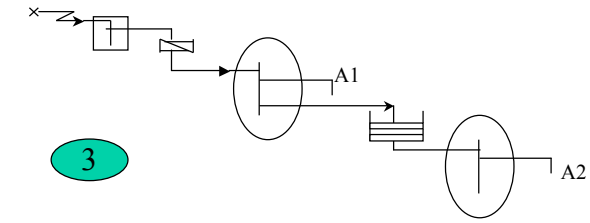
Ordonnancement des actions (1)

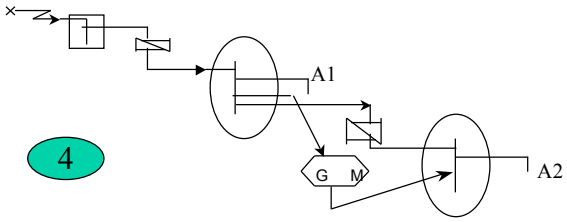


jean-philippe.babau@insa-lyon.fr

Tâches Vs actions (2)

- 2 actions en séquence A1 – A2



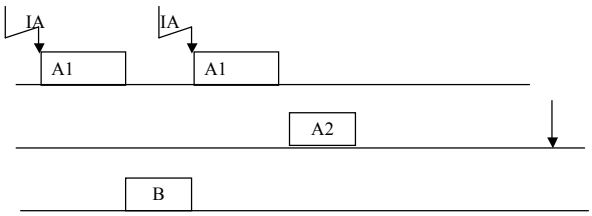


jean-philippe.babau@insa-lyon.fr

Ordonnancement des actions (2)

3

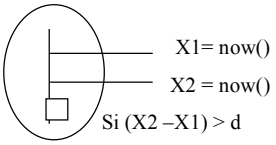
4



jean-philippe.babau@insa-lyon.fr

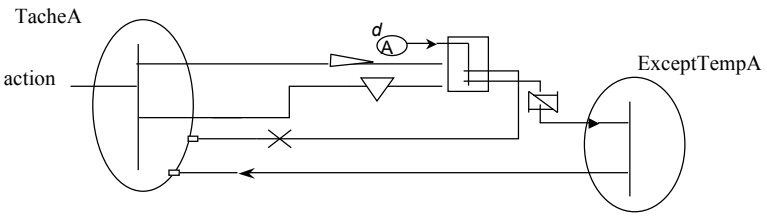
Suivi du respect des échéances

- Instrumentation temporelle des tâches



➔ trop tard !

- Activation d'une tâche corrective

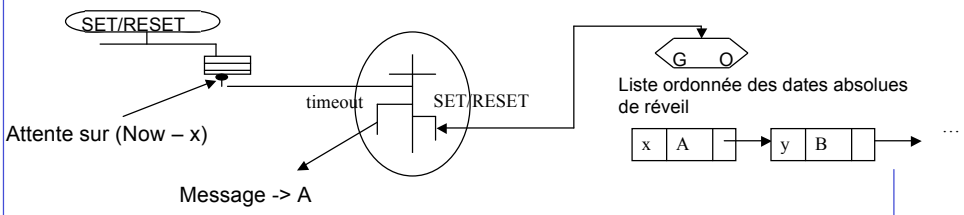


jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Suivi du respect des échéances

- Gestion groupée des timeout



- Set
  - ajout dans la liste
  - les paramètres sont la date de réveil et l'adresse de la bal où envoyer le message si expiration
- Reset
  - retrait de la liste
- Attente sur la date du prochain réveil
- Expiration
  - Émission du message

jean-philippe.babau@insa-lyon.fr

Découpage de l'application

- Approche modulaire
  - Encapsulation
  - Réutilisation
- Découpage en phases
  - Plusieurs processus distincts
    - Lien entre les processus
      - Événement déclenchant et événement de fin
        - » Détection et réaction prioritaire (urgence)
      - suspension / activation des activités en cours / nouvelles
        - » Gestion du cycle de vie des activités du système
    - Partage des données
  - Plusieurs modes de fonctionnement par tâches
    - Mode : information partagée
    - Pas de préemption

jean-philippe.babau@insa-lyon.fr

Découpage de l'application

- Processus serveurs
  - Une tâche offre des services aux autres tâches
    - Modèle de communication (cf. serveur ci après)
  - Problème de gestion de la concurrence d'accès
    - Inversion de priorité possible pour un appel bloquant
      - Cf. cours ordonnancement
    - Priorité des services pour des événements concurrents
      - File et priorité
    - Rendre la priorité haute : par exemple pour les services I/O
    - Rendre la tâche moins prioritaire : traitements sans délai

jean-philippe.babau@insa-lyon.fr

Définition des tâches

- Approche événementielle
  - Une tâche est mise en place pour une séquence d'action liée à un événement externe et possédant une priorité liée à l'urgence temporelle de l'événement
- Approche service
  - Une tâche par processus
  - Une tâche pour un ensemble de services
- Approche système
  - Une tâche à haute priorité pour exécuter des services de haut niveau non fournis par le RTOS

jean-philippe.babau@insa-lyon.fr

Echange d'informations

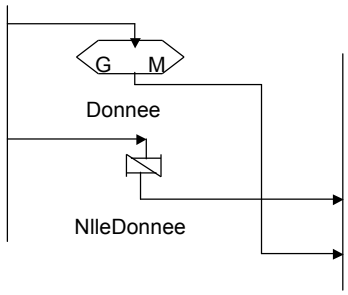
- Messages
  - Besoins
    - Requêtes
    - Invocation distante
    - Serveurs
    - Objets
  - Mise en oeuvre
    - Boite aux lettres
    - Canal
- Données partagées
  - Variables globales
  - Fichiers
  - Mémoire partagée

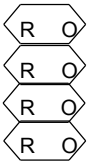
jean-philippe.babau@insa-lyon.fr

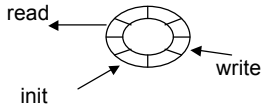
Département Informatique INSA de Lyon

Echange d'informations

- Données synchronisées
  - Ecrasement
  - Nouvelle valeur
  - Échange désynchronisé
- Nombre maximal fixe de données
  - buffer matériel (réseau)
  - 1 à n section critiques



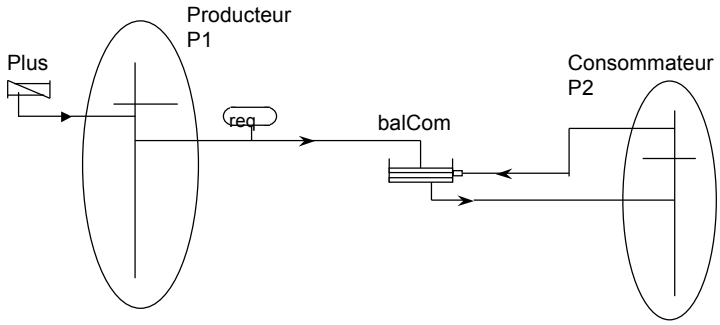




jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Boîte aux lettres (LACATRE)



jean-philippe.babau@insa-lyon.fr

Boîte aux lettres (C/VxWorks)

```
LOCAL MSG_Q_IDbalComId;

typedef struct
{       int num;
        char message[5];
        } info;

void Producteur()
{
int i = 0 ;
info msgE;
for (;;)
{   semTake(PlusId,WAIT_FOREVER);
    msgE.num = i ++;
    msgE.message =  " toto";
    msgQSend(balComId,(char*)&msgE, sizeof(info), WAIT_FOREVER,
        MSG_PRI_NORMAL) ;
}}
```

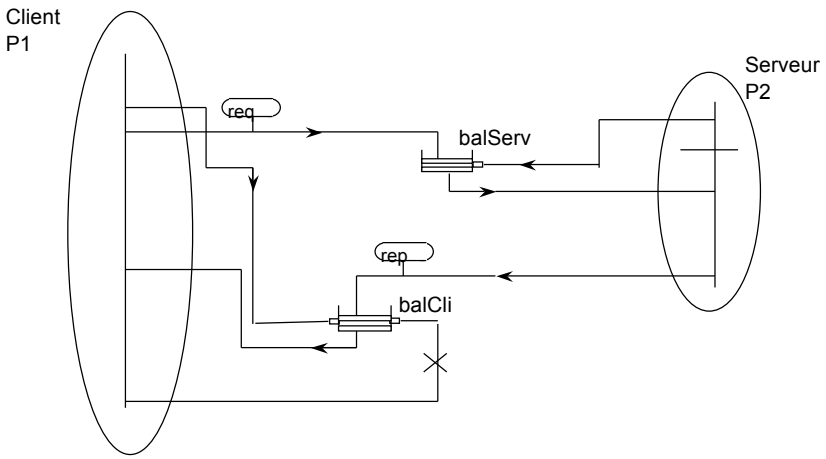

Boîte aux lettres (C/VxWorks)

```
void Consommateur()
{
info msgR ;

balComId = msgQCreate(10,sizeof(info),MSG_Q_FIFO);
for ( ; ; )
{       msgQReceive(balComId ,(char*)&msgR,sizeof(info),WAIT_FOREVER);
        printf("%i%s\n",msgR.num,msgR.message);
}}
```

Communication Client / Serveur

- Le serveur crée une bal de réception des requêtes
- Le client envoie une requête dans la bal de réception des requêtes
  - mode d'adressage ou de nommage
- Si réponse nécessaire
  - création d'une bal de réponse
  - envoi avec la requête de la bal de réponse
  - attente de la réponse

Communication Client/Serveur (LACATRE)



Communication client / serveur (VxWorks)

```
LOCAL MSG_Q_ID  balServId;
LOCAL MSG_Q_ID  balCliId;

typedef struct  {info message ;
                 MSG_Q_ID balRepId ;  }MSG;

void Client()
{
MSG  reqC;
MSG  repC;

balCliId = msgQCreate(10,sizeof(MSG),MSG_Q_FIFO);

reqC.message = … ;
reqC. balRepId = balCliId ;

msgQSend(balServId,(char*)& reqC, sizeof(MSG), WAIT_FOREVER, MSG_PRI_NORMAL) ;
msgQReceive(balCliId ,(char*)& repC,sizeof(MSG),WAIT_FOREVER);

}
```

jean-philippe.babau@insa-lyon.fr

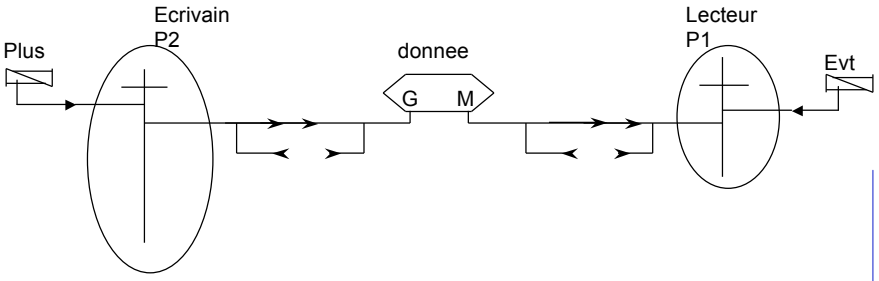
Département Informatique INSA de Lyon

Communication client / serveur (VxWorks)

```
void Serveur()
{
MSG  reqS;
MSG  repS;

balServId = msgQCreate(10,sizeof(MSG),MSG_Q_PRIORITY);
for ( ; ; )
{    msgQReceive(balServId ,(char*)& reqS,sizeof(info),WAIT_FOREVER);
      repS.message = … ;
      repS. balRepId = NULL ;
     msgQSend(reqS.balRepId,(char*)& repS, sizeof(MSG), WAIT_FOREVER,
                MSG_PRI_NORMAL) ;
}}
```

jean-philippe.babau@insa-lyon.fr

Donnée partagée (LACATRE)



Donnée partagée (VxWorks)

```
typedef struct
{
        int heure ;
        int minute ;
} horaire;

horaire donnee ;

semDonneeId = semMCreate(SEM_Q_PRIORITY | SEM_INVERSION_SAFE) ;
...

void Lecteur()
{
semTake(EvtId, WAIT_FOREVER) ;
    semTake(semDonneeId, WAIT_FOREVER) ;
        printf("%i %i\n", donnee.heure, donnee.minute ) ;
    semGive(semDonneeId) ;
}
```

Donnée partagée (VxWorks)

```
void Ecrivainr()
{
semTake(PlusId, WAIT_FOREVER) ;

    semTake(semDonneeId, WAIT_FOREVER) ;
        if (donnee.minute < 59 )
        {     donnee.minute ++ ;}
        else
        {     if (donnee.heure < 23)
              {     donnee.heure ++ ;
                    donnee.minute = 0 ; }
              else
              {     donnee.heure = 0 ;
                    donnee.minute = 0 ; }
        }
    semGive(semDonneeId) ;
}
```

jean-philippe.babau@insa-lyon.fr

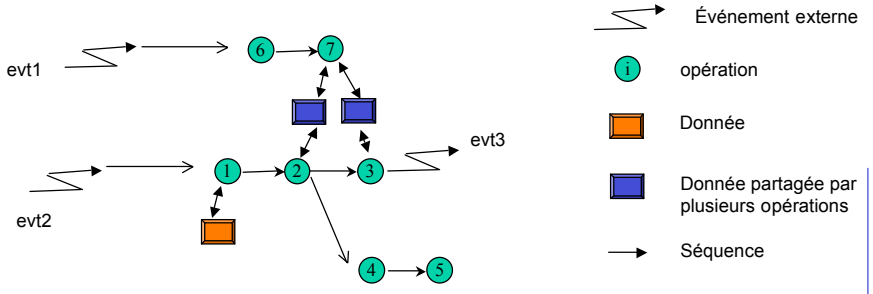
Echanges de données

- Zones de stockage
  - Taille d'une zone de stockage (producteur /consommateur)
  - Politique de stockage
    - FIFO, LIFO, politique d'écrasement
  - Datation d'acquisition des données stockées
  - Validité des données : age, valeurs correctes
- Du point de vue de l'écrivain
  - Si la zone de mémorisation est pleine
    - Perte ou attente
    - Perte de la donnée la plus récente ou de la plus ancienne
    - Mise en attente (timeout)
  - Evénement lié à l'écriture
- Du point de vue du lecteur
  - Toutes les données doivent être reçues
  - Dernière(s) donnée(s) reçue(s)
  - Attente si pas de données, attente avec timeout

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

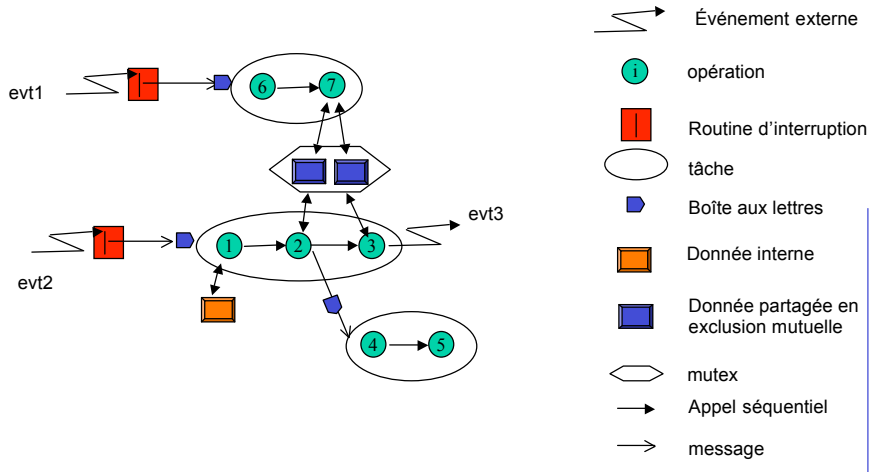
Exemple



jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

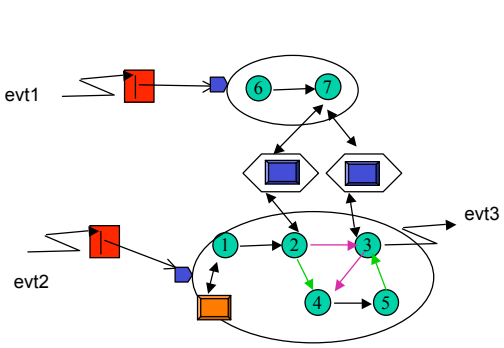
Exemple



jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Exemple

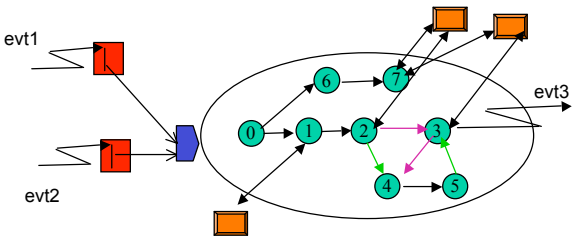


Événement externe
opération
Routine d'interruption
tâche
Boîte aux lettres
Donnée interne
Donnée partagée en exclusion mutuelle
mutex
Appel séquentiel
message

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Exemple



Événement externe
opération
Routine d'interruption
tâche
Boîte aux lettres
Donnée interne
Donnée partagée en exclusion mutuelle
mutex
Appel séquentiel
message

jean-philippe.babau@insa-lyon.fr

Création / destruction

- Principe de base : accès à un objet initialisé (créé)
- Gestion statique (main)
  - Ordre de création
    - objets de communication
    - tâches
    - IT, alarmes
  - Ordre de destruction
    - Ordre inverse
- Gestion dynamique
  - tâche « propriétaire »
  - « la tâche crée sa bal »
- Gestion pseudo-dynamique
  - Création statique d'un pool
  - Création dynamique dans la limite du pool

jean-philippe.babau@insa-lyon.fr

Statique Vs dynamique

- Statique
  - prédictible, pire cas
  - dimensionnement
  - coûteux
- Dynamique
  - optimisation
  - adaptation
    - environnement inconnu
    - changement de mode
    - mode dégradé
- Semi dynamique
  - allocation limitée à une borne supérieure
- Statique / dynamique
  - application programmée en statique
  - superviseur dynamique

jean-philippe.babau@insa-lyon.fr

Vérification

- Preuves
  - Absence d'interblocage
  - Dimensionnement correct des éléments de communication
  - Analyse temps réel
- Tests
  - Test unitaire des fonctions
  - Test d'intégration
    - Enchaînement d'actions, tâches
  - Scénarios
    - Combinaison réaliste d'événements
      - E1 E1 E2
      - E1 E2 E1 …
      - Prise en compte des contraintes temps réel
    - Comportement estimé de l'environnement

Mise au point

- Environnement externe
  - Ensemble complexe
    - avion
  - Coûteux
    - Quelles sont les conséquences sur le procédé si échec du test ?
  - Non réalisé lors de la mise au point du logiciel
  - Environnement simulé ou émulé
- Différentes versions
  - Simulation
    - IT + routine = tâche
  - Émulation logicielle
    - Une tâche-procédé envoie une IT
  - Le test modifie le logiciel par ajout d'éléments
    - Test de l'ensemble « appli+module de test »
  - Émulation matérielle
    - Module externe spécifique
  - Environnement réel

Mise au point

- Mode trace
  - Fonction, temps
  - Ressources restreintes (IHM, BD, medium de communication
  - Mars PathFinder
- Modularité
  - Attention aux effets de bord (Ariane V)
  - Intégration

jean-philippe.babau@insa-lyon.fr

Conclusion

- Formalisme graphique LACATRE
  - Simple pour la communication
  - Complet pour l'implémentation
- Architecture
  - Scrutation / événementiel
  - Concurrence
    - Contrôle dynamique des actions
    - Contraintes de temps
  - Informations échangées
    - Synchronisation
    - Avec/sans écrasement
    - Taille de la zone de stockage
  - Statique / dynamique

jean-philippe.babau@insa-lyon.fr

Conclusion

- Gestion du temps
  - Indépendante des traitements
- Vérification
  - Sûreté de fonctionnement
  - Tests sur combinaison d'événements
  - Environnement externe

jean-philippe.babau@insa-lyon.fr