

Plan

- 1 Le modèle standard
 - Approche événementielle
 - Causalité
 - Abstraction d'un calcul
- 2 Clichés (snapshots)
 - Prise de cliché
 - Utilisation des clichés
- 3 Description des algorithmes
 - Description du comportement des processus
 - Exemple : l'élection



Principes algorithmiques

- Algorithmes symétriques
 - code répliqué,
 - données initiales propres : identité, voisinage de communication.
- Structurer les échanges de messages :
 - réseau en anneau
 - arbre
 - maillage (graphe complet)
- Étudier des problèmes génériques :
 - Les services : datation, exclusion mutuelle, consensus, élection. . .
 - Les observations de propriétés stables : terminaison, interblocage
 - La tolérance aux fautes : réplication, atomicité

[Précis 2.2.5, 2.2.6 pp.32–34]



Description des algorithmes



```
Process P(id : 0..N-1)
  <variables locales>
  on <condition-logique> :
    <Action>
  on reception message(arg) [ from P(j) ] :
    <Action>
  on <condition-logique> ^ reception message(arg) :
    <Action>
  on start : // initialement
    <Action>
  ...
```

- Action : modification des variables locales et/ou envoi(s) de message, ou terminaison (**terminate**)
- Envoi : **send** Msg(<args>) **to** <destinataire(s)>
- Choix d'un événement à traiter : non déterministe parmi ceux ayant la garde vraie et un message à consommer

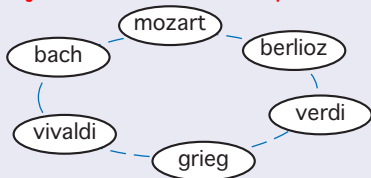


Exemple : l'élection



Le problème de l'élection

Objectif : Élire un seul processus



- Un processus a une identité unique qu'il connaît
- Un processus ne connaît pas le nombre global de processus
- Un processus ne connaît pas l'identité des autres
- Communication sur un anneau logique

[\[Précis 2.2.9 pp.39-41 \]](#)

1. *An improved algorithm for decentralized extrema-finding in circular configurations of processes*, Ernest Chang and Rosemary Roberts. Communications of the ACM, May 1979.



Solution correcte ou fausse ?



On suppose que les processus sont totalement ordonnés (ici par leur indice, en pratique, par leur adresse IP par exemple)

```
Process P(id : 0..N-1)
  //  $\ominus$  et  $\oplus$  : opérateurs modulo  $N$ 
  type Etat = {candidat, élu};
  Etat étatCourant ← candidat;
  on reception Candidat(proc) from P[id $\ominus$ 1] :
    if (proc < id) send Candidat(proc) to P[id $\oplus$ 1];
    else if (proc = id) étatCourant ← élu;
    else nop; // ignorer le message
  on (étatCourant = élu) :
    terminate;
```

Pourquoi cela ne marche-t-il pas ?



Solution qui conduit à l'élection du plus petit



```
Process P(id : 0..N-1)
  type Etat = {candidat, élu};
  Etat étatCourant ← candidat;
  on start:
    send Candidat(id) to P[id⊕1]; // chacun candidate
  on reception Candidat(proc) from P[id⊖1]:
    if (proc < id) send Candidat(proc) to P[id⊕1];
    else if (proc = id) étatCourant ← élu;
    else nop; // ignorer le message
  on (étatCourant = élu) :
    terminate;
```

Pas parfait : un seul processus se termine



Solution plus complète : tous les processus terminent

```
Process P(id :0..N-1) {
  type Etat = {candidat,élu,perdant};
  Etat étatCourant ← candidat;
  on start :
    send Candidat(id) to P[id⊕1]; // chacun candidate
  on reception Candidat(proc) from P[id⊖1]:
    if (proc < id) send Candidat(proc) to P[id⊕1];
    else if (proc = id) étatCourant ← élu;
    else nop; // ignorer le message
  on (étatCourant = élu) :
    send Elu(id) to P[id⊕1];
  on reception Elu(proc) from P[id⊖1]:
    if (proc ≠ id) then
      étatCourant ← perdant;
      send Elu(proc) to P[id⊕1];
  endif
  terminate
```

Déclenchement spontané individuel



Pas nécessairement tous candidats au départ (mais tous éligibles)

```
Process P(id : 0..N-1)
  type Etat = {candidat, élu, perdant};
  Etat étatCourant ← candidat;
  on random() :
    send Candidat(id) to P[id⊕1];
  on reception Candidat(proc) from P[id⊖1]:
    if (proc < id) send Candidat(proc) to P[id⊕1];
    else if (proc = id) étatCourant ← élu;
    else if (proc > id) send Candidat(id) to P[id⊕1];
  :
```



Conclusion

- Modélisation par des **événements** locaux
- Relation entre ces événements, en particulier la **causalité**
- Représentation avec des chronogrammes
- Notion d'**état global**, de **coupure**
- Calcul d'un état global

