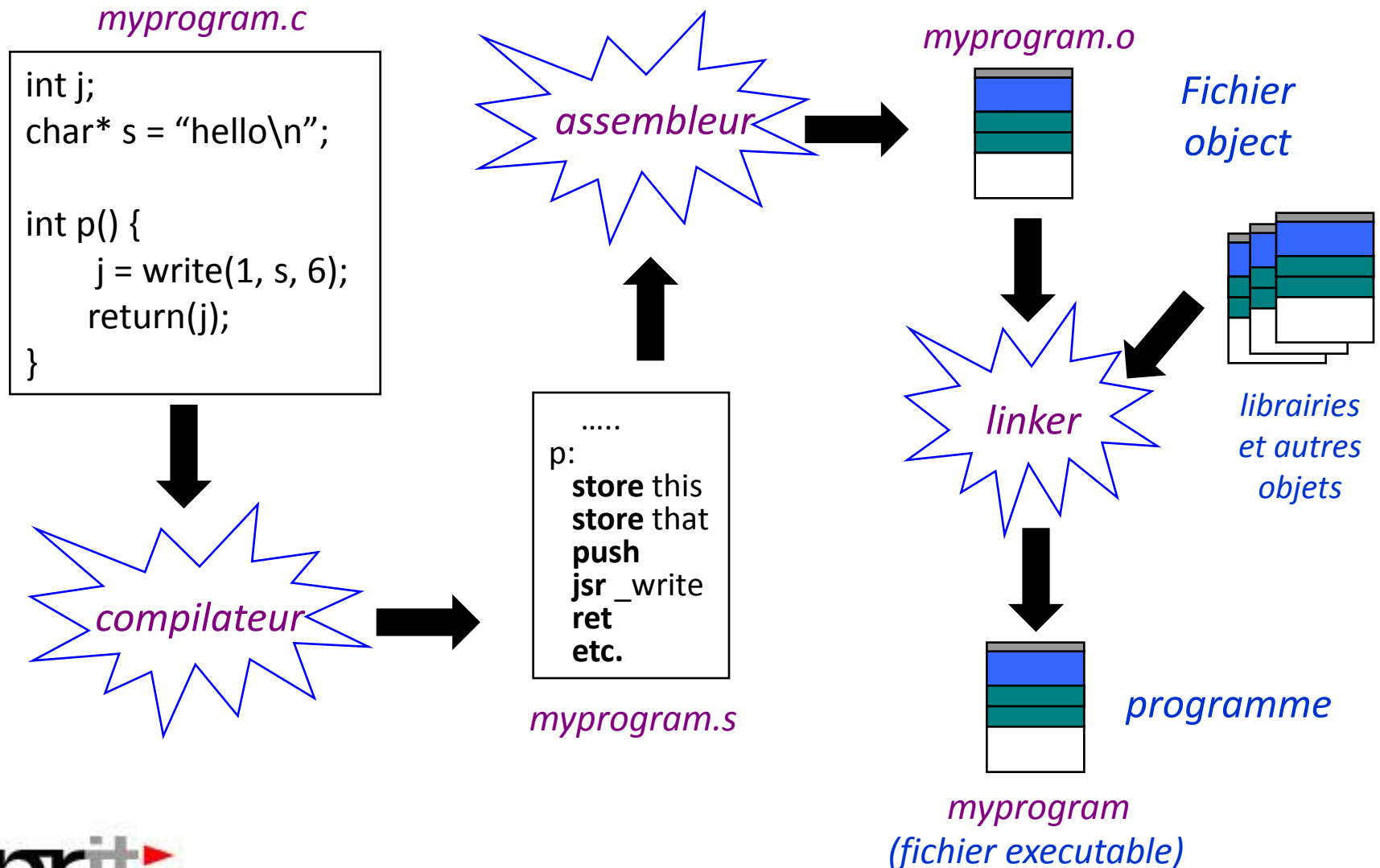


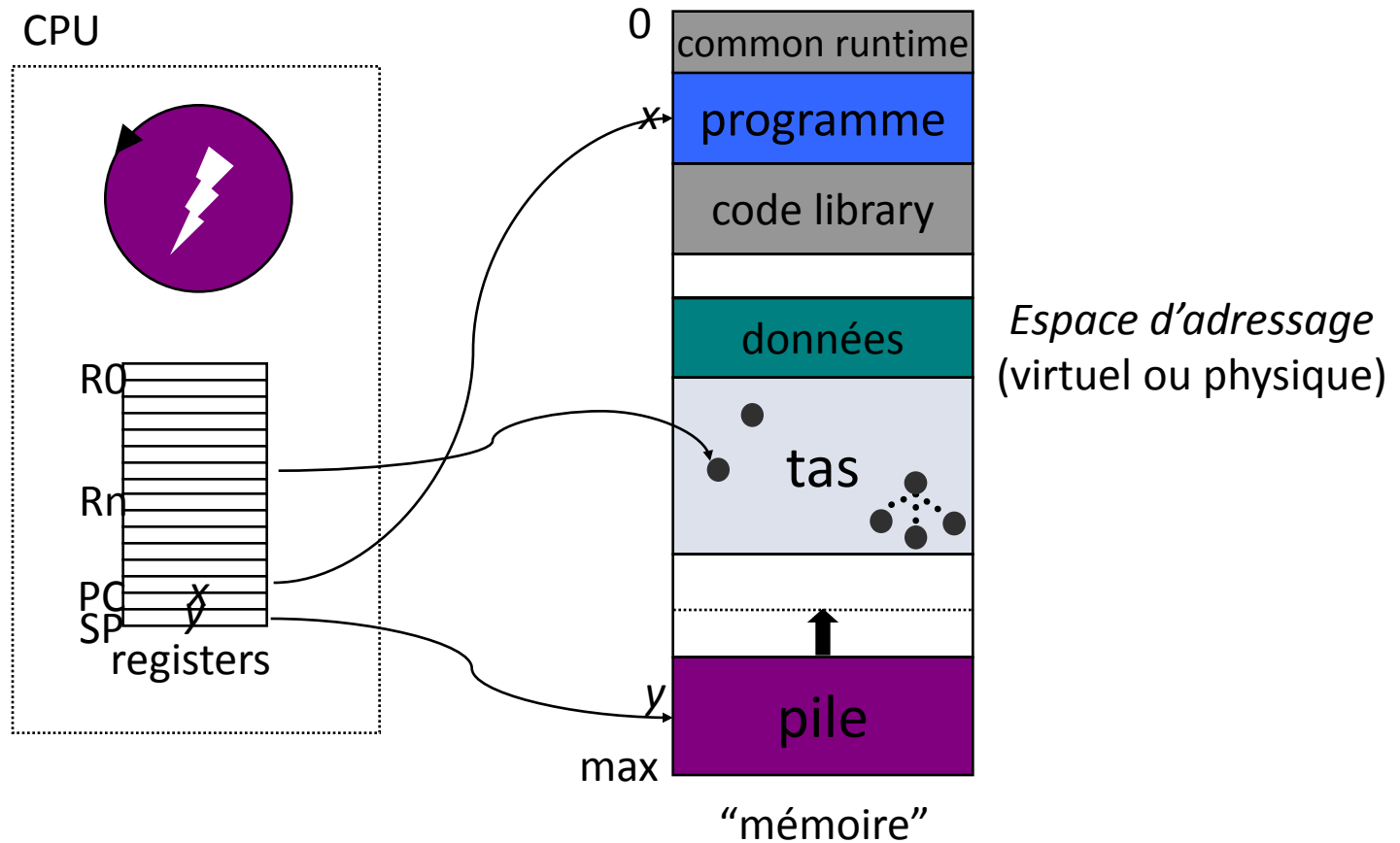
Cours n°2: Gestion des processus

**SYSTÈMES D'EXPLOITATION  
AVANCÉES**

# La naissance d'un processus...



# ...A son exécution



# Définissons...

---



- Un programme est composé d'une suite d'instructions qui agissent sur un ensemble de données : **objet statique**
- **Processus** : lorsque l'on déclenche l'exécution d'un programme, il devient un **objet dynamique** géré par l'OS.
- Un **processus** est composé d'un **programme** et de l'ensemble des **ressources** reliées à l'exécution du programme. Ces ressources incluent de la mémoire, des I/Os, des fichiers ouverts par le programme, du temps de CPU et autres.

→ **Processus = unité d'exécution (unité de partage du temps processeur et de la mémoire) ≠ Programme**

# Système d'exploitation multitâches

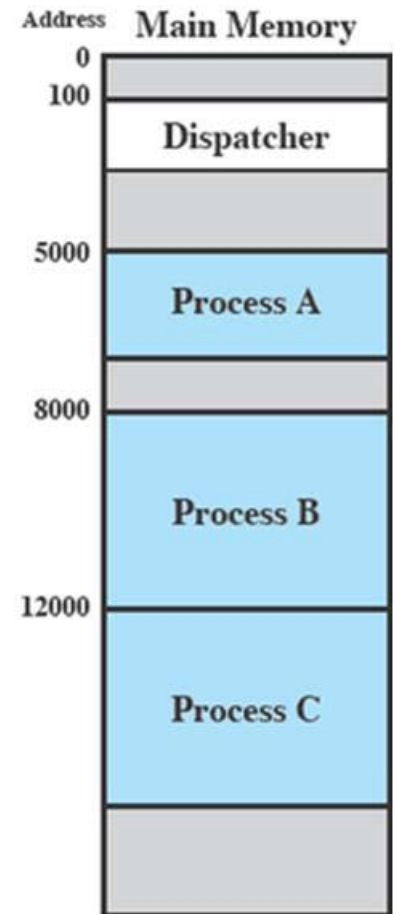


- Un système d'exploitation est dit *multitâches* s'il permet d'exécuter, apparemment simultanément, plusieurs programmes.
- Ce fonctionnement est réalisé en alternant rapidement l'exécution de différents processus c'est-à-dire en effectuant un *multiplexage temporel* du processeur (chaque processus s'exécute pendant une fraction de seconde)
- L'exécution des processus est *entremêlée*
- Par conséquent, il ne s'agit pas réellement d'un traitement simultané sauf si la machine a plusieurs processeurs

# Processus et Multiprogrammation



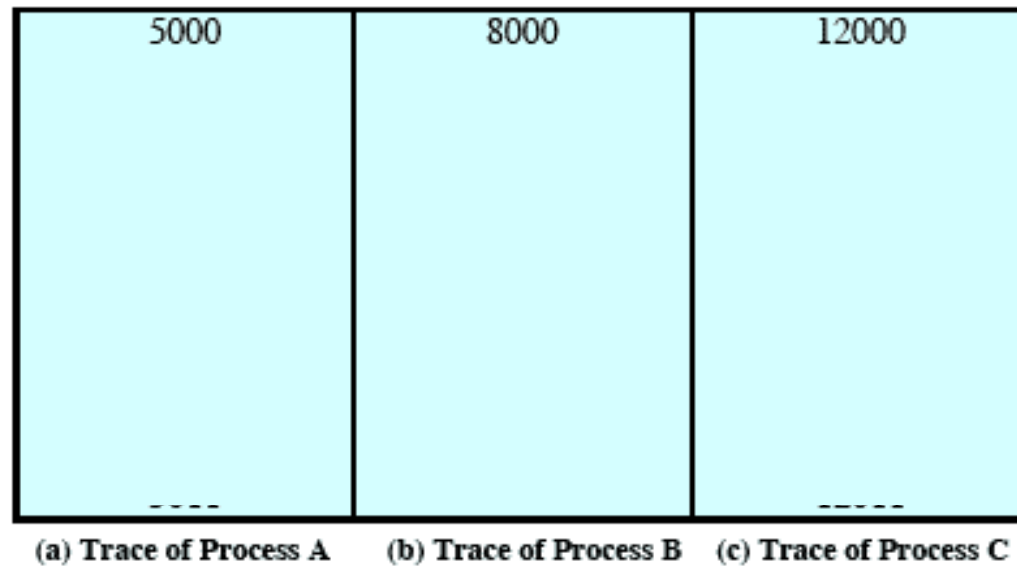
- Soient 3 processus A, B et C à exécuter
- A, B et C sont chargés simultanément en mémoire (avec le Dispatcher)
- Le comportement d'un processus donné = séquence d'instructions qu'il est en train d'exécuter (***Trace***)
- Le ***Dispatcher*** est un programme qui commute le processeur d'un processus à un autre



# Trace du point de vue des processus

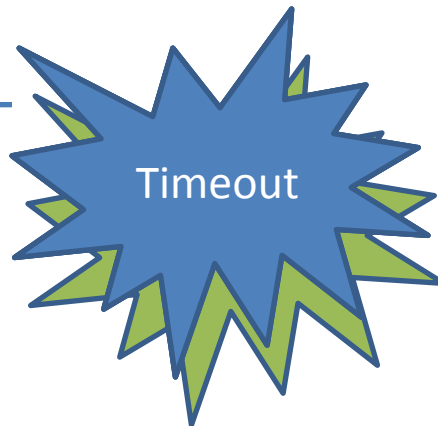
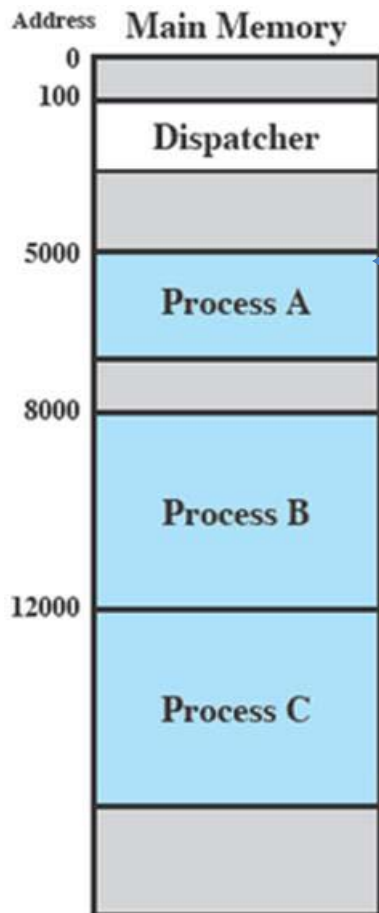


- Chaque processus s'exécute jusqu'à la fin de sa tâche



5000 = Starting address of program of Process A  
8000 = Starting address of program of Process B  
12000 = Starting address of program of Process C

# Trace du point de vue du processeur



1	5000
2	5001
3	5002
4	5003
5	5004
6	5005

Timeout

7	100
8	101
9	102
10	103
11	104
12	105

13	8000
14	8001
15	8002
16	8003

I/O Request

17	100
18	101
19	102
20	103
21	104
22	105

23	12000
24	12001
25	12002
26	12003

27	12004
28	12005

Timeout

29	100
30	101
31	102
32	103
33	104
34	105

35	5006
36	5007
37	5008
38	5009
39	5010
40	5011

Timeout

41	100
42	101
43	102
44	103
45	104
46	105

47	12006
48	12007
49	12008
50	12009
51	12010
52	12011

Timeout

100 = Starting address of dispatcher program

Shaded areas indicate execution of dispatcher process;  
first and third columns count instruction cycles;  
second and fourth columns show address of instruction being executed



# Feuille de route

---

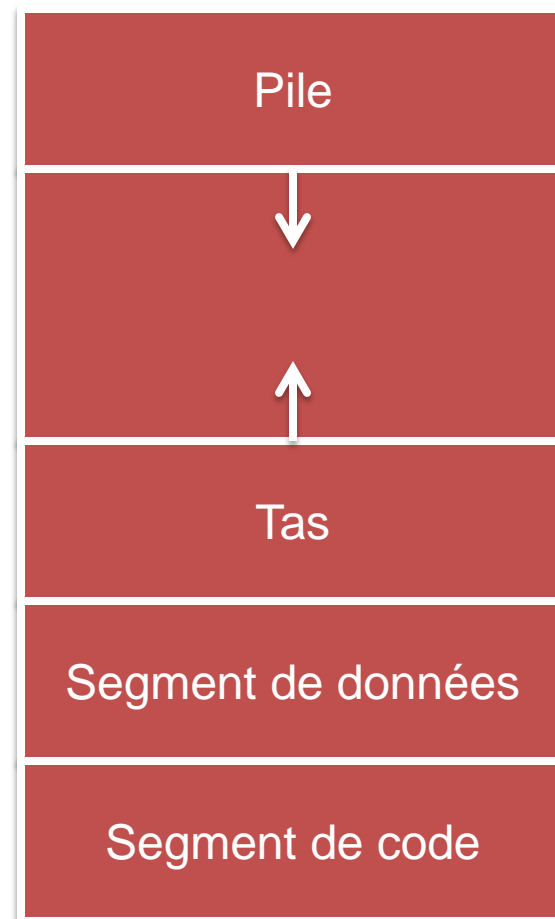


- Comment les processus sont-ils représentés et contrôlés par l'OS?
- Quels sont les états caractérisant le comportement des processus?
- Quels sont les structures de données utilisées pour gérer les processus?
- Comment l'OS utilise-t-il ces structures de données pour contrôler l'exécution des processus?

# Structure de l'espace mémoire d'un processus



- Espace mémoire dont la structure est définie par l'OS (Ensemble des adresses mémoire que le processus peut adresser)
- Plusieurs zones =
  - Segment de code (**text** section)
    - Copie du segment de code du fichier exécutable
  - Compteur de programme
    - Indique la prochaine instruction à exécuter
  - Contenu des registres du processeur
  - Pile d'exécution
    - Paramètres des fonctions, variables locales, adresses de retour, pointeur de pile, etc.
  - Segment de données (**data** section)
    - variables globales (.bss and .data).
  - Un Tas
    - Pour la mémoire allouée dynamiquement

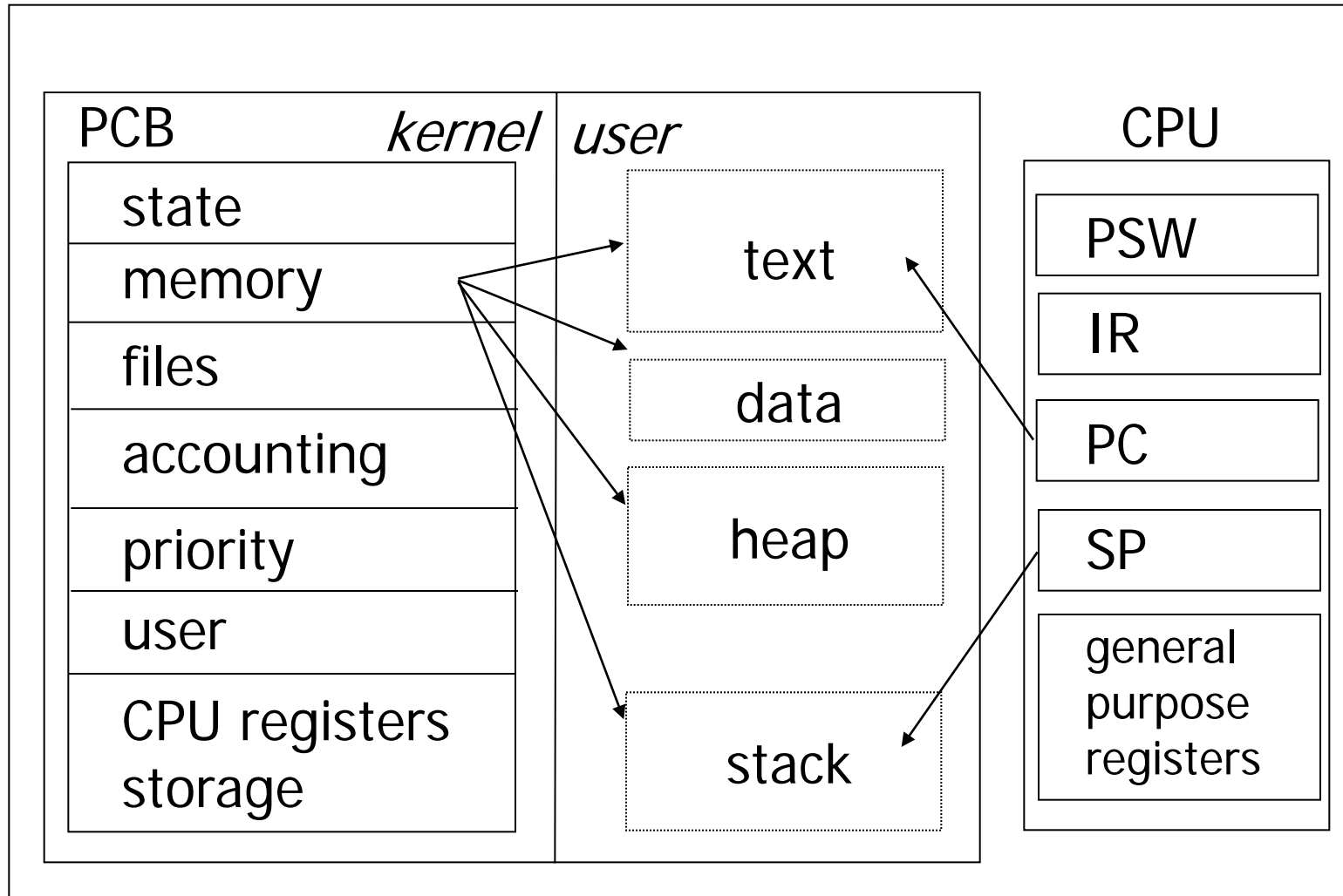


# Bloc de contrôle de processus (PCB)

- Identifiants numériques
  - Identifiant du processus
  - Identifiant du processus qui a créé ce processus (processus père)
  - Identifiant de l'utilisateur
- Informations sur l'état des processus
- Compteur de programme (@ de la prochaine instruction à exécuter)
- Contenu des registres du CPU
- Information d'ordonnancement
  - Priorité,...
- Information de gestion de la mémoire
  - Valeur du registre de base et limite,...
- Information comptable
  - Taux d'utilisation du processeur,...
- Etats des dispositifs d'E/S
  - E/S alloués, liste des fichiers ouverts,...

pointeur	état de processus
numéro de processus	
compteur programme	
registres	
limites mémoire	
liste des fichiers ouverts	
⋮	

# Bloc de contrôle de processus



# Feuille de route

---



- Comment les processus sont-ils représentés et contrôlés par l'OS?
- Quels sont les états caractérisant le comportement des processus?
- Quels sont les structures de données utilisées pour gérer les processus?
- Comment l'OS utilise-t-il ces structures de données pour contrôler l'exécution des processus?

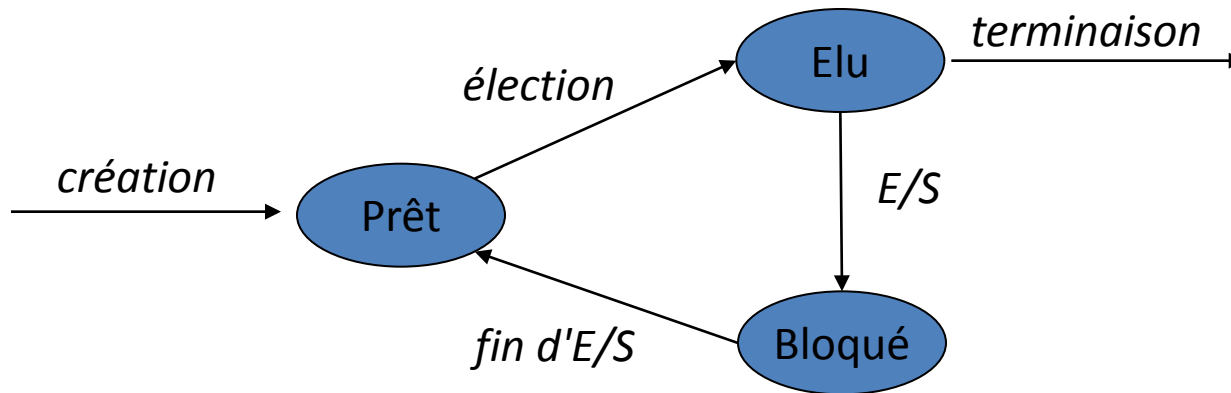
# Multiprogrammation



## Mode batch

Le processus actif rend la main :

- lorsqu'il se termine
- lorsqu'il se bloque en attente d'une E/S



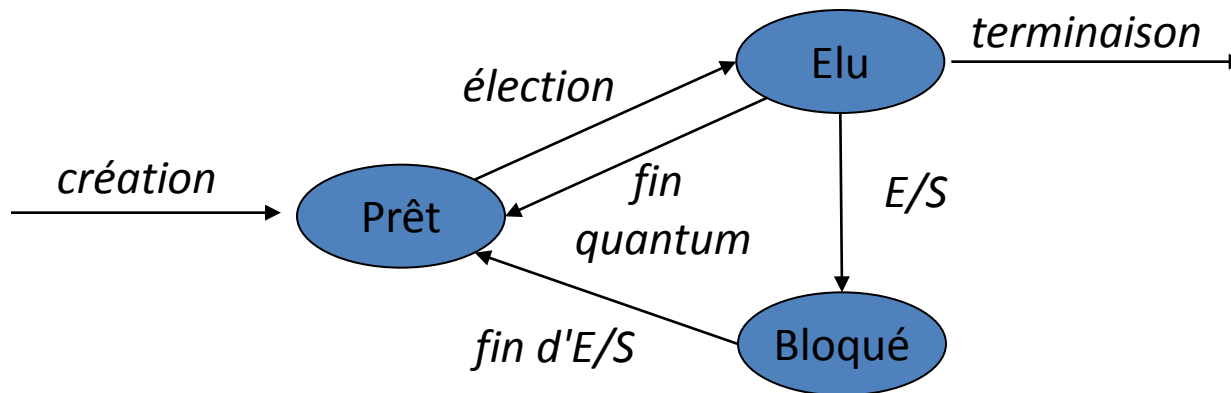
# Temps partagé



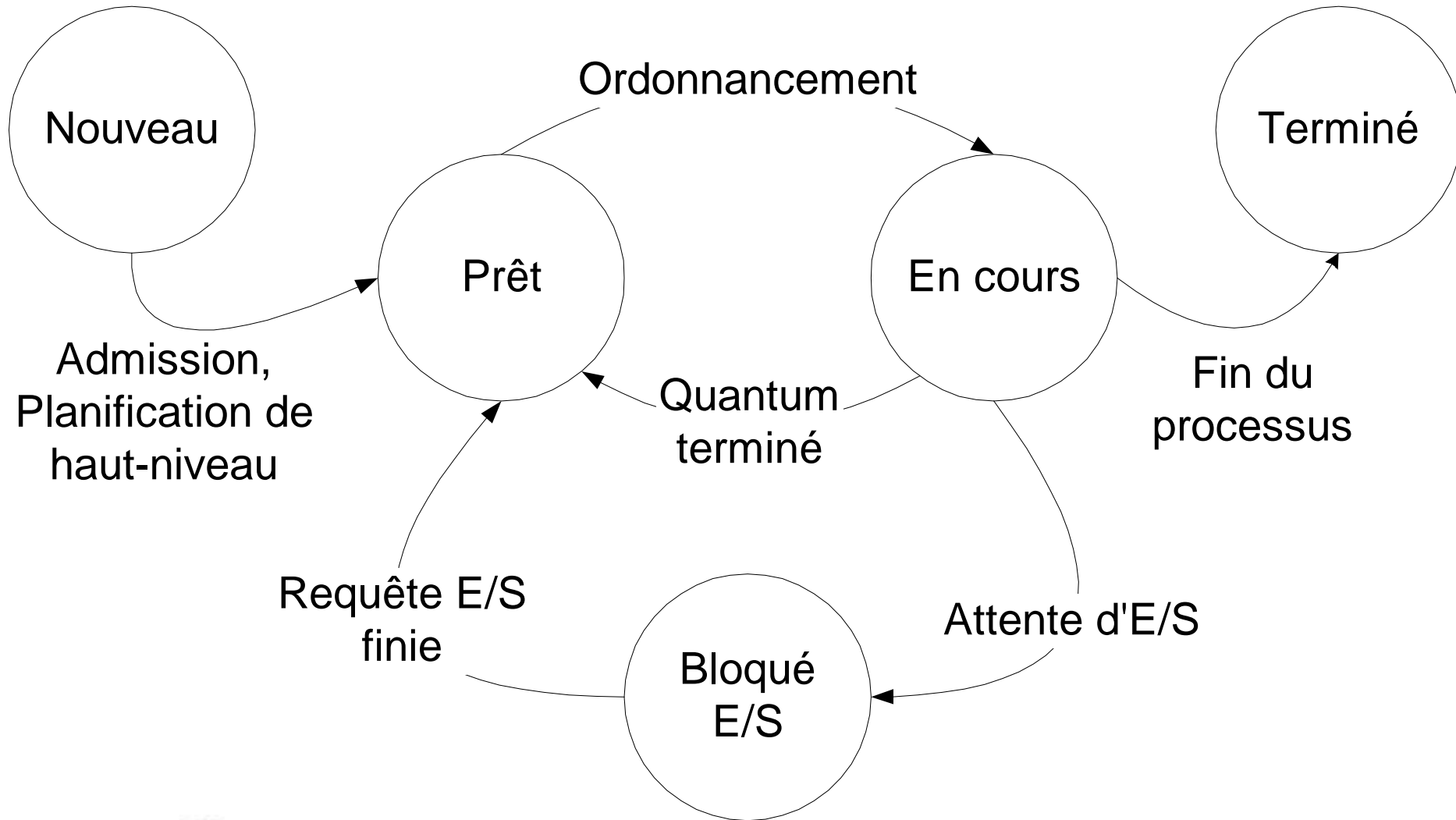
## Temps partagé

Le processus actif rend la main :

- lorsqu'il se termine
- lorsqu'il se bloque en attente d'une E/S
- lorsqu'il a épuisé son quantum de temps



# Etats d'un processus





# Création de processus: Pourquoi?

---



- Initialisation du système
  - Exécution d'un appel système de création de processus par un processus en cours d'exécution
  - Requête utilisateur sollicitant la création d'un nouveau processus
  - Lancement d'un travail en traitement par lot
- Un nouveau processus est créé du fait qu'un processus existant exécute un appel système de création de processus

# Création de processus: Comment?

---



- Affecter un identifiant unique (pid)
- Allouer de l'espace mémoire pour l'image du processus (code, données, pile...)
- Initialiser le bloc de contrôle de processus (valeurs par défaut, état = New, pas de dispositifs d'E/S ou de fichiers ouverts...)
- Mettre en œuvre les liens appropriés (ajouter le nouveau PCB à la liste chaînée utilisée pour les queues d'ordonnancement)

# La fin d'un processus

---



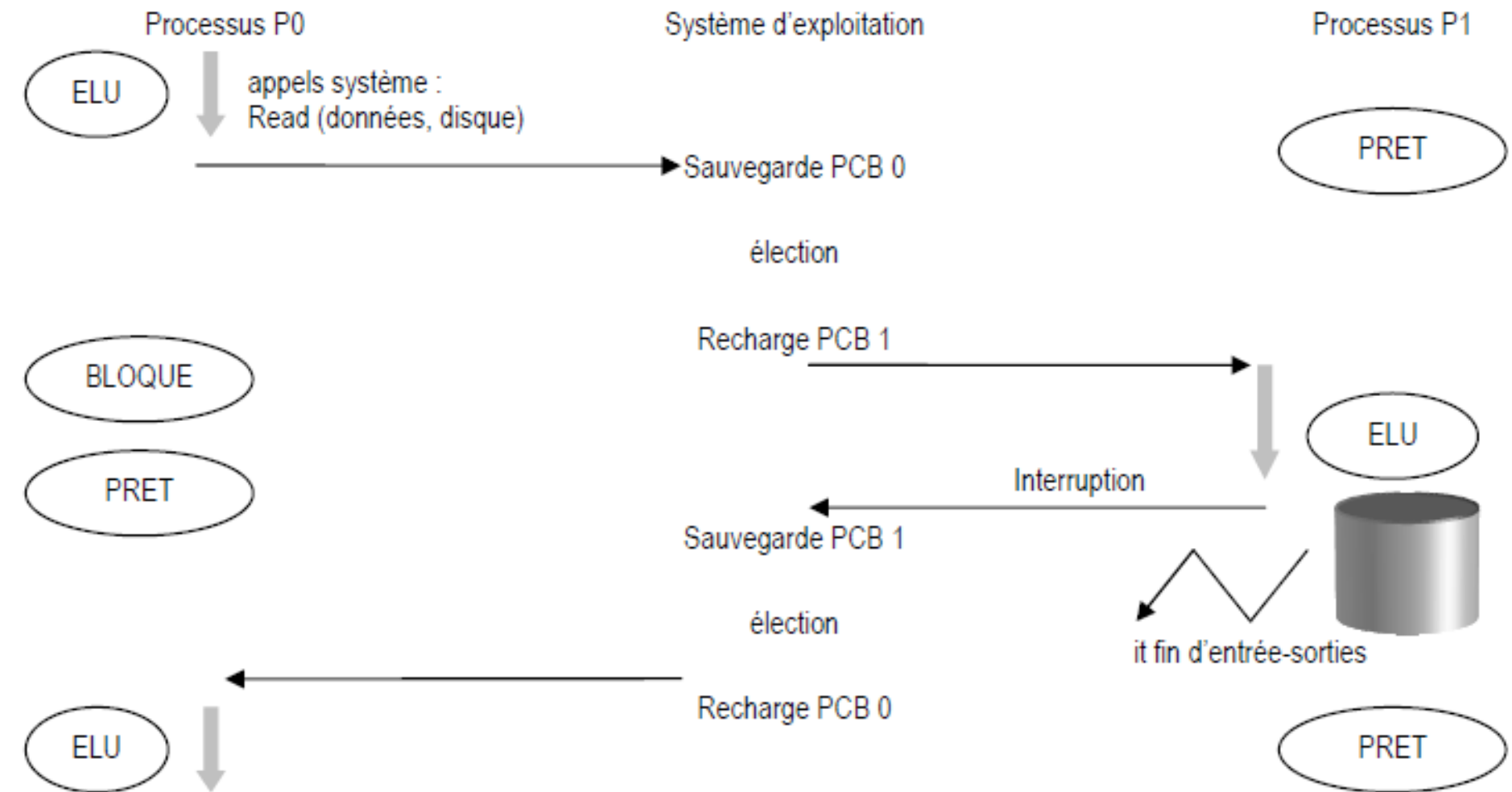
- Arrêt normal (volontaire)
- Arrêt pour erreur (volontaire)
- Arrêt pour erreur fatale (involontaire)
- Le processus est arrêté par un autre processus (involontaire)

# Commutation du CPU : Comment?



- Sauvegarder le contexte du CPU (PC et autres registres)
- Mettre à jour le PCB du processus en cours d'exécution avec le nouvel état et autre infos
- Changer le PCB dans la file d'attente appropriée (prêts, bloqués, ...)
- Choisir un autre processus à exécuter
- Mettre à jour le PCB du processus sélectionné (état: en cours d'exécution)
- Restaurer l'état du CPU à partir du PCB du processus sélectionné

# Commutation entre processus



# Feuille de route

---



- Comment les processus sont-ils représentés et contrôlés par l'OS?
- Quels sont les états caractérisant le comportement des processus?
- Quels sont les structures de données utilisées pour gérer les processus?
- Comment l'OS utilise-t-il ces structures de données pour contrôler l'exécution des processus?

# Les entités responsables de l'ordonnancement

