



Unité d'enseignement Systèmes d'exploitation centralisés

1ère Année Informatique et Mathématiques Appliquées

TD : Signaux

30 mars 2018

### Thèmes traités

- Notions de base et mise en œuvre des signaux UNIX.
- Opérations essentielles de l'API signaux Unix, protocole d'usage.
- Programmation d'horloges
- Sauvegarde et restauration de points de reprise (API et protocole d'usage)

## 1 Questions

1. Qu'apporte le mécanisme d'interruptions à la gestion et la supervision des E/S par le système d'exploitation ? Est-ce que l'utilisation de ce mécanisme est pertinente dans tous les cas ? Pourquoi ?
2. Du point de vue de l'application et du point de vue des mécanismes mis en jeu, quelles sont les différences entre la réception d'un signal et l'appel d'une procédure ?

## 2 Déroulement

- présentation du mécanisme de signaux, mise en œuvre du masque, signaux masqués et pendants (planches 3-5 des transparents d'accompagnement)
- revue rapide des opérations de base (**signal/sigaction**, **kill**, **pause**, **alarm**) (planches 6,8 à 10)  
schéma de base de mise en place d'un traitant (planche 7)  
Exemple : retour sur l'ordonnanceur, avec le traitant de **SIGALRM**
- exercice 2.5.3 sans timers
- revue rapide des opérations de manipulation des masques de signaux (**sigsuspend/sigprocmask/sigpending**) (planche 10)
- présentation des timers (planche 14). Exercice 2.5.3 avec timers
- présentation de **setjmp/longjmp**.  
Illustration avec le principe de mise en œuvre des exceptions (planche 13)

# Signaux Unix

Version Logicielle du mécanisme d'interruption API SX

↳ idée : service de programmation | événementielle  
réactive

↳ ensemble d'événements

↳ pour chaque événement définir une réaction associée

S'abonner (evt, réaction)

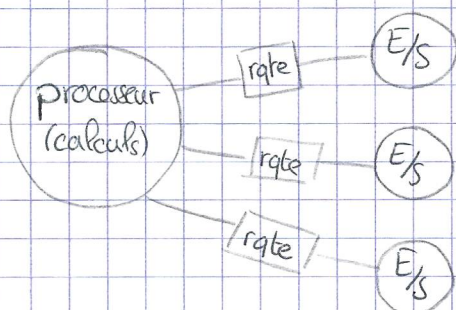
Publier (evt)

↳ procédure exécutée à chaque réception d'un evt.

## I / Questions

1. / But : Tous les processeurs fonctionnent en parallèle.

↳ Détecter les périphériques inactifs.

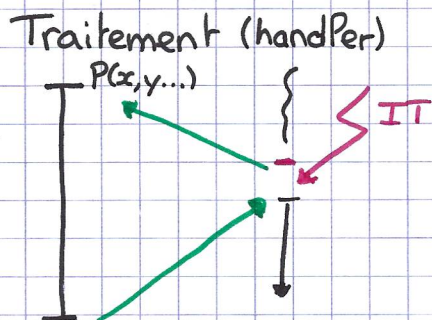


→ processus de scrutation / attente active.  
⊖ coûteux en tps processeur.

→ interruptions : le périphérique prévient de la fin d'une E/S.

⊕ efficace, réaction immédiate,  
libère le processeur de la scrutation.  
⊖ matériel spécial.

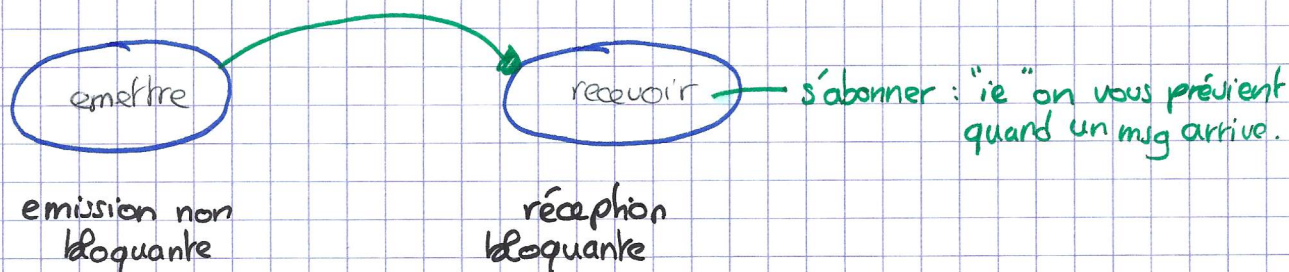
2. / Mécanisme d'interruption



Dans un programme classique, on choisit du temps to d'exécution du programme.

Dans le mécanisme d'exécution par interruption, le sspgm peut être exécuté à n'importe quel moment.

## II / Déroulement





La primitive alarm(to) demande l'émission d'un signal SIGALRM dans to sec.

Exercice 2.5.3 :

```
static int nbrecus = 0;  
static int nbalarm = 0;
```

```
void affsig (int sig)
```

```
    printf ("Recepto d'un signal %d \n", sig);  
    nbrecus++;
```

```
}
```

```
void achif (int sig)
```

```
    printf ("Reception du signal %d (SIGALRM)\n", sig);  
    printf ("Toujours achif...\n");  
    alarm(1);  
    nbalarm++;
```

```
}
```

```
int main() {
```

```
    int i;
```

```
    for (i = 1; i <= NSIG; i++)
```

```
        signal(i, affsig);
```

```
    signal(SIGALRM, achif);
```

```
    alarm(3);
```

```
    while ((nbrecus != 5) && (nbalarm != 9)) {
```

```
        pause();
```

```
}
```

```
    printf ("reçu %d alarm %d \n", nbrecus, nbalarm);  
    return 0;
```

```
}
```