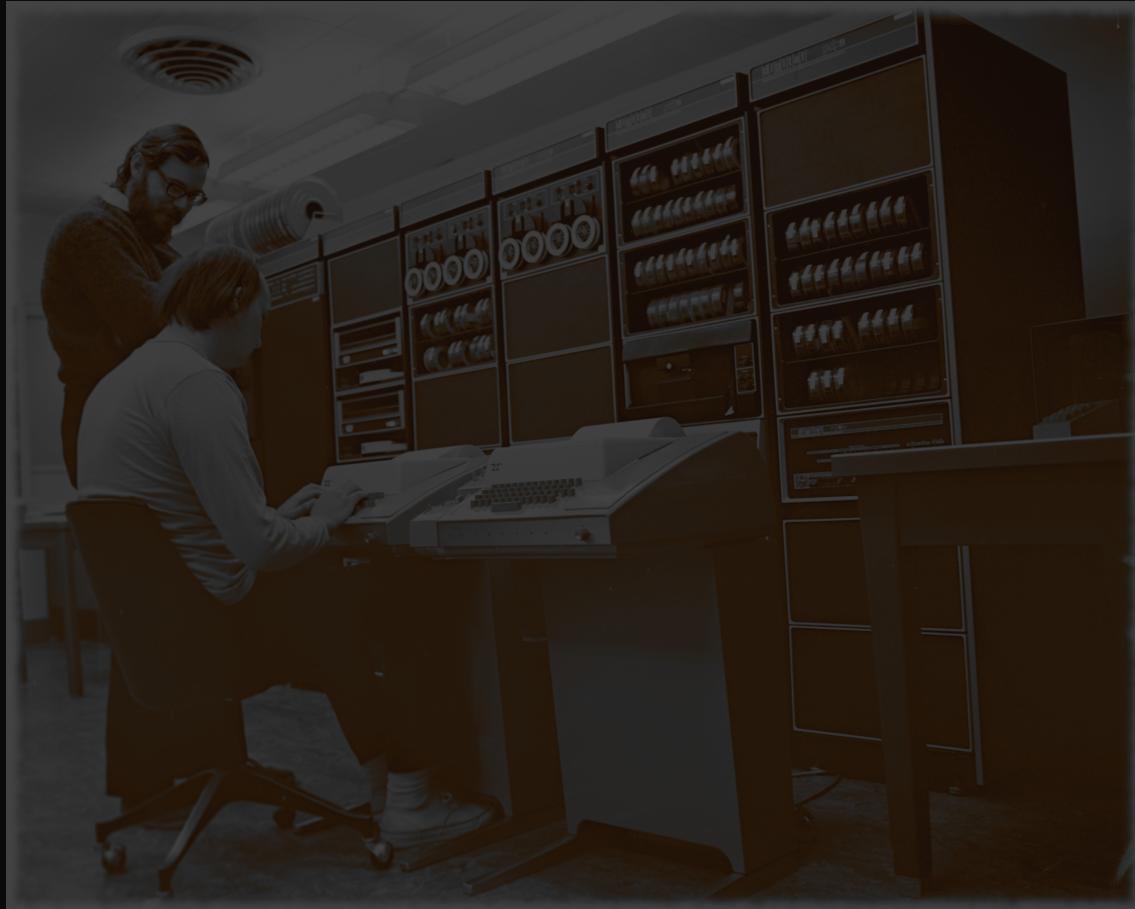


# Systèmes d'exploitation

TP : criblage par processus filtrants

Alain Lebret

2024-2025



**Objectif**

Appréhender la communication interprocessus par tubes anonymes en mettant en oeuvre un pipeline.

**Pré-requis** : chapitre sur la communication interprocessus – section tubes

**Durée estimée** : 1 séance

## Présentation

En 1978, Charles Antony Richard Hoare a proposé de calculer les nombres premiers inférieurs à  $N^2$  en utilisant  $N + 2$  processus qui reçoivent de leur prédécesseur une suite croissante de nombres dont le premier,  $n$ , est un nombre premier, et transmettent à leur successeur les nombres de cette suite, dans le même ordre, à l'exception de  $n$  et de ses multiples.

Nous nous proposons d'utiliser des tubes de communication afin de permettre aux processus de s'échanger les suites de nombres, puis de sauvegarder, en fin de compte, dans le fichier « prime\_numbers.txt », l'ensemble des nombres premiers inférieurs ou égaux à une valeur passée en argument au programme (`max_val`).

Pour ce faire, nous mettons en place le mécanisme suivant (voir figure 2)

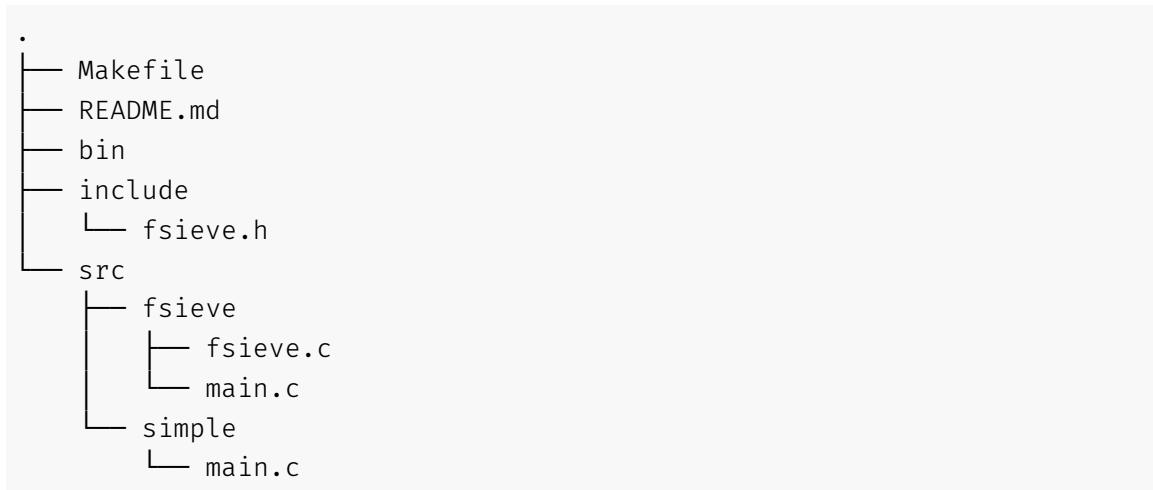
Un père va créer un fils  $f_1$  puis va lui communiquer la suite des entiers compris entre 2 et `max_val` à l'aide d'un tube anonyme. De manière générale, un fils  $f_i$  reçoit comme premier entier un nombre  $n_i$  qui est premier et qu'il sauvegarde dans le fichier. Il peut alors filtrer les entiers qui suivent :

- si l'entier est divisible par  $n_i$  il ne fait rien ;
- au premier entier  $n_i+1$  non divisible par  $n_i$  il crée lui-même un processus fils ainsi qu'un tube de communication avec celui-ci ; l'entier  $n_i+1$  ainsi que tous les autres entiers non divisibles par  $n_i$  sont alors envoyés à son fils.

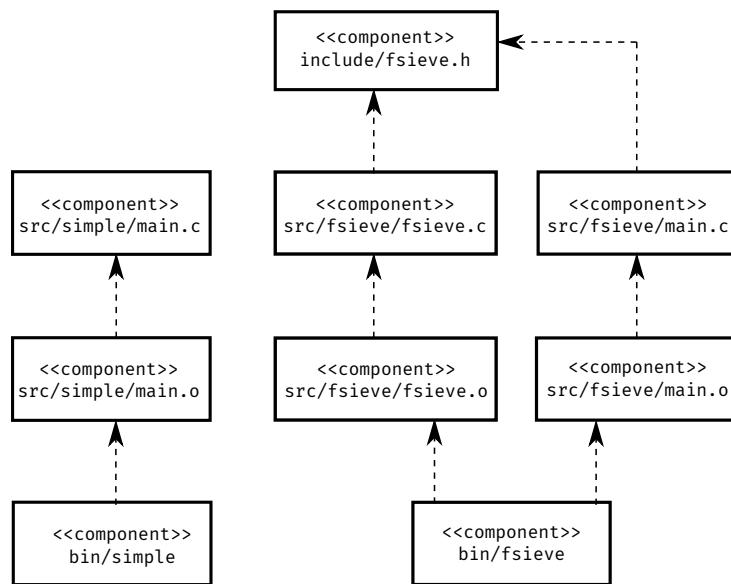
Lorsque le père a terminé d'envoyer tous ses entiers, il envoie l'entier -1 à son fils puis attend la mort de ce dernier.

## Travail à réaliser

1. Créez le sous-dossier « fsieve » dans votre dossier « os » puis dans ce dernier, créez la structure arborescente suivante :



2. Réalisez le fichier « Makefile » de façon à ce qu'il respecte le diagramme de composants de la figure 1.



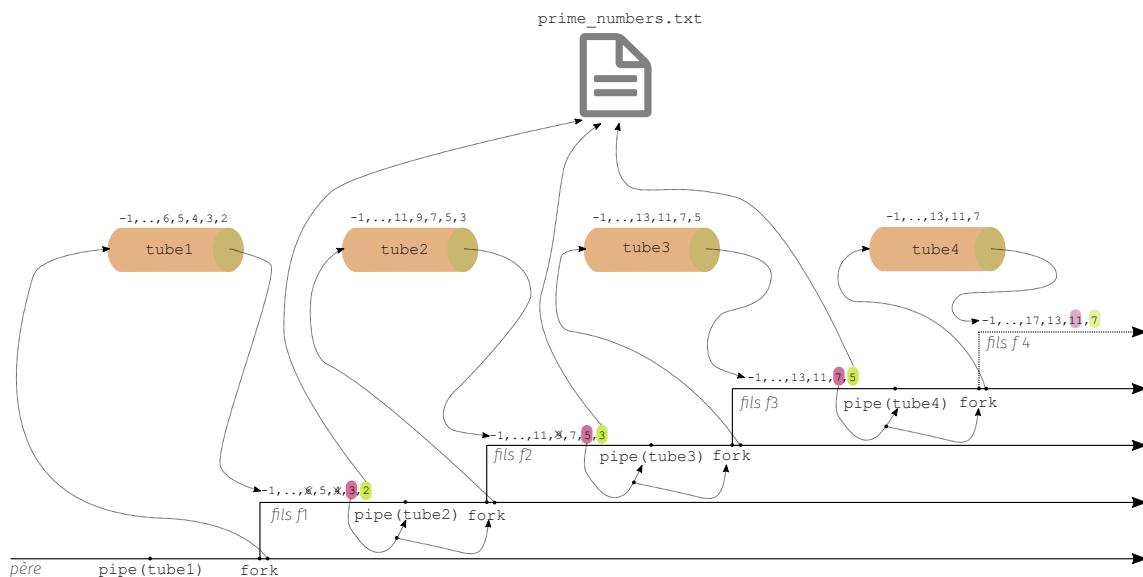
**Figure 1:** Diagramme de composants.

### Exercice n°1

1. Implémentez le programme « simple » dans lequel un processus père envoie à son fils, par l'intermédiaire d'un tube anonyme, une suite de nombres entiers que le fils doit afficher sur sa sortie standard. La réception du nombre -1 termine le fils.
2. Compilez et testez le programme.

## Exercice n°2

1. Codez le programme « fsieve » qui met en oeuvre un pipeline permettant à chaque processus fils de celui-ci d'extraire un nombre premier spécifique et de filtrer tous ses multiples (voir figure 2).
2. Compilez et testez le programme.



**Figure 2:** Représentation du début du mécanisme.



Le fichier « prime\_numbers.txt » étant partagé par le père et ses fils, il sera nécessaire de faire appel à la fonction `fflush()` après chaque écriture dans le fichier.

## Livrable

Avant la date limite, déposez sur <https://foad.ensicaen.fr> l'archive compressée « `fsieve-.zip` » qui contiendra le contenu de tout le dossier « `fsieve` », excepté les fichiers objets et les exécutables.

Le fichier « `README.md` » permettra de :

- vous identifier ;
- décrire brièvement le contenu du dossier (objectif et composants) ;
- donner les instructions pour compiler à l'aide de la commande **make** les programmes.

## Résumé

Durant cette séance vous avez réalisé un pipeline de calcul en mettant en oeuvre une communication par tubes anonymes entre vos processus.