

# Programmation système

## Description du projet

J'ai implémenté une primitive de communication entre processus comme demandé dans l'énoncé, toutes les fonctions demandées ont été implémentées en respectant les contraintes d'atomicité.

## Structure du conduit

```
struct conduct{
    const char *name;
    size_t c;
    size_t a;
    void * buffer;
    int eof;
    int nbElement;
    pthread_mutex_t verrou;
    pthread_cond_t lectureOk;
    pthread_cond_t ecritureOk;
    int debut;
    int fin;
};
```

Concernant la structure de donnée j'ai choisi d'utiliser un tampon circulaire, l'entier *debut* de la structure *conduct* est un index indiquant le début des données dans le tampon, et l'entier *fin* indique la fin des données.

## Comparaison avec les tubes

J'ai comparé mon conduit au pipe, grâce au programme suivant :  
on a des threads writers qui envoient la structure suivante à des threads readers :

```
typedef struct demande
{
    long writerId;
    int x;
    int y;
```

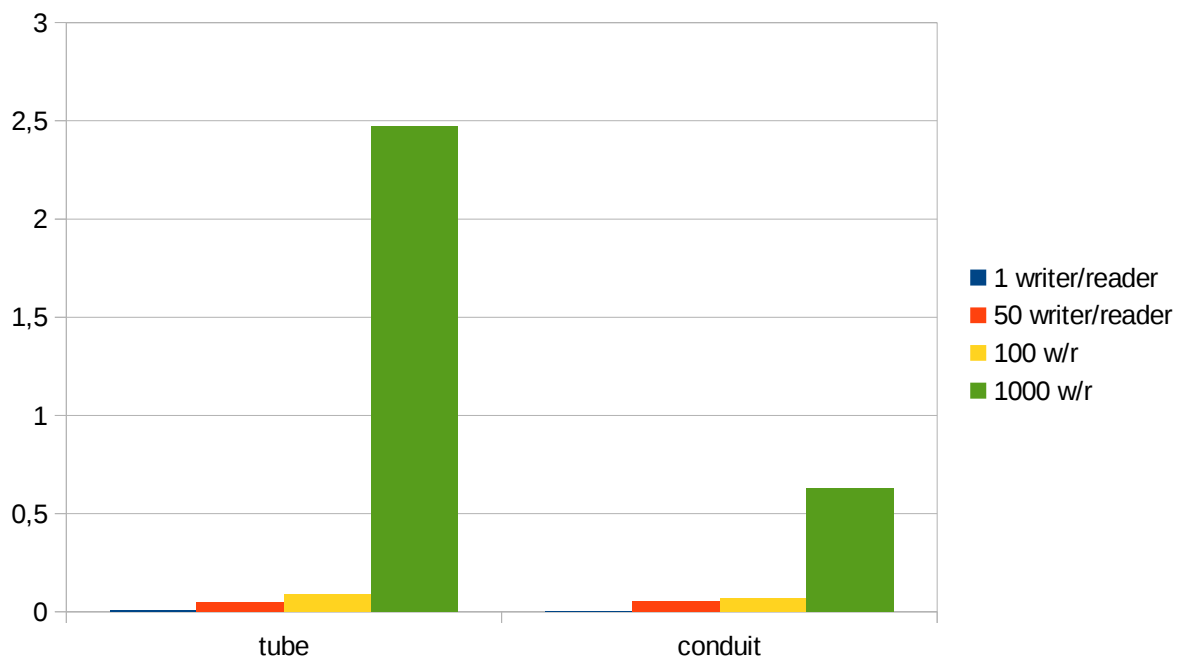
}demande;

Lorsque les threads readers reçoivent la demande ils calculent le pgcd de x et y puis le renvoie à un écrivain(pas forcément celui qui à envoyé la demande) grâce à la structure suivante :

```
typedef struct resultat
{
    struct demande * dmd;
    long readerId;
    int res;
}resultat;
```

Les tests on été effectués sur une machine possédant deux cœurs, un processeur AMD E-450 et 4GO de mémoire vive.  
Les temps d'exécutions ont été mesurées grâce à la commande time.

Voici les résultats obtenues :



	1 writer/reader	50 writer/reader	100 w/r	1000 w/r
tube	0,005	0,049	0,091	2,472
conduit	0,004	0,052	0,070	0,63