



GETPID

```

pid_t get_pid(void);
  
```

output :
pid_t -> le numero du processus au format pid_t (format numerique qu'on peut stocker en int)

input :
void

KILL

```

int kill(pid_t pid, int sig);
  
```

output :
int -> 0 si pas d'erreur, -1 si erreur

input :
pid_t pid -> le pid du processus auquel on veut envoyer le signal
int sig -> le signal qu'on veut envoyer

SIGACTION

```

int sigaction(int signum, const struct sigaction * __Nullable restrict act, struct sigaction * __Nullable restrict oldact);
  
```

output :
int -> 0 si pas d'erreur, -1 si erreur

input :
int signum -> le numero du signal que l'on veut intercepter
const struct sigaction * __Nullable restrict act -> pointeur sur une structure qui définit comment gérer le signal, par ex: dans ce programme on va configurer cette structure pour exécuter une fonction quand on reçoit un signal SIGUSR1 ou SIGUSR2 pour gérer la réception d'un bit 0 ou 1 et recevoir un char de 8 bits.
struct sigaction * __Nullable restrict oldact -> pointeur sur un sigaction, permet d'avoir une sauvegarde de l'ancien comportement pour le restaurer: par ex: on veut remettre le traitement de signal par défaut après un ctrl-c

PID

PID or process ID is a decimal number used to identify each process on a computer. the function getpid() unless it returns this number.