

Coquéran Logan

1.1)

La fonction `socket()` est bien un appel système car, par exemple, dans le man, on trouve sa documentation dans le 2ème chapitre, celui correspondant aux appels système.

1.3)

```
#include <stdio.h>
```

```
#include <sys/socket.h>
```

1.4)

```
int main(const int argc, const char **argv)v
```

1.5)

```
sock = socket(AF_INET, SOCK_STREAM, 0)
```

AF_INET : Protocole internet IPv4

SOCK_STREAM : Mode connecté, connection dans les deux sens

0 : On laisse le noyau choisir le protocole

1.6)

On obtient toujours 3 pour la valeur de la prise. Cette valeur est toujours la même car on pointe vers le même point de communication.

2.1)

```
struct sockaddr_in{
    short  sin_family;
    u_short sin_port;
    struct in_addr sin_addr;
    char    sin_zero[8];
};
```

2.2)

`sin_family` désigne le type de connection que l'on va réaliser, pour avoir un "abonnement internet", il faut mettre cette valeur à AF_INET

2.3)

`sin_addr` désigne l'adresse ip à laquelle on va se connecter. Si l'on veut se connecter localement on va mettre cette valeur à 127.0.0.1 (lo)

2.3)

`sin_port` désigne le port de l'application par lequel on va passer. On le met à `htons(APP_PORT)` pour avoir un port sélectionné localement.

2.4)

2.5)

2.6)

2.7)

2.8)

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <string.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <arpa/inet.h>

int main(const int argc, const char **argv)
{
    int sock;
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0 )
    {
        perror("erreur de socket");
        return 1;
    } else {
        printf("Le socket à bien été créer. Descripteur : %d\n", sock);
    }

    struct sockaddr_in s_addr;

    s_addr.sin_family = AF_INET;
    s_addr.sin_addr.s_addr = INADDR_ANY;
    s_addr.sin_port = htons(APP_PORT);

    memset(&(s_addr.sin_zero), 0, 8);

    if (bind(sock, (struct sockaddr_in *) &s_addr, sizeof(s_addr)) < 0)
    {
        perror("bind error");
        return 1;
    } else {
        printf("server binded");
    }

    if (listen(sock, BACKLOG) < 0) {
        perror("listen error");
    }
}
```

```
        return 1;
    } else {
        printf("Server listening\n");
    }

    return 0;
}
```