

Aide-Mémoire – Programmation en C

Cette aide-mémoire rappelle les fonctions de manipulation des sockets et des types de données pour la communication réseau. Seulement le prototype est spécifié ici. Si vous souhaitez connaître le retour de ces méthodes, merci de consulter le man sur votre terminal (`man nom_fonction`).

1. Structures de données

- Adresse Internet :

```
#include <netinet/in.h>

struct in_addr {
    uint32_t s_addr;
};
```

- Caractéristiques IP d'un socket :

```
#include <netinet/in.h>

struct sockaddr_in {
    sa_family_t sin_family; /* address family: AF_INET */
    in_port_t sin_port; /* port in network byte order */
    struct in_addr sin_addr; /* internet address */
};
```

- Caractéristiques d'une machine hôte :

```
#include <netdb.h>

struct hostent {
    char *h_name; /* official name of host */
    char **h_aliases; /* alias list */
    int h_addrtype; /* host address type */
    int h_length; /* length of address */
    char **h_addr_list; /* list of addresses */
};

#define h_addr h_addr_list[0] /* First host address */
```

2. Fonctions de manipulation des structures de données

- Représentation des nombres standards :

```
#include <arpa/inet.h>

uint32_t htonl(uint32_t hostlong); /* host to network ip conversion */
uint16_t htons(uint16_t hostshort); /* host to network port conversion */
uint32_t ntohl(uint32_t netlong); /* network to host ip conversion */
```



```
uint16_t ntohs(uint16_t netshort); /* network to host port conversion */
```

- Manipulation des adresses :

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
char *inet_ntoa(struct in_addr in); /* convert ip address to a string in
                                     IPv4 dotted-decimal notation */
```

- Manipulation des noms de machine :

```
#include <netdb.h>
struct hostent *gethostbyname(const char *name); /* return host information
                                                  for the given host name
                                                  */
```

3. Fonctions de manipulation des sockets TCP

- Création d'un socket :

```
#include <sys/types.h>
#include <sys/socket.h>
int socket(
    int domain,      /* AF_UNIX, AF_INET, ... */
    int type,        /* SOCK_DGRAM, SOCK_STREAM, ... */
    int protocol     /* 0 protocole par défaut */
);
```

- Fermeture d'un socket :

```
#include <unistd.h>
int close(int fd);
```

- Attachement d'un socket à une adresse :

```
#include <sys/types.h>
#include <sys/socket.h>
int bind(
    int sockfd,          /* descripteur de la socket à attacher */
    const struct sockaddr *name, /* pointeur sur l'adresse à utiliser */
    socklen_t addrlen    /* Longueur de l'adresse */
);
```

- Ouverture du service :

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



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

```
int listen(
```

```
    int sockfd,      /* socket de rendez-vous */
```

```
    int backlog      /* nombre maximum de connexions pendantes */
```

```
);
```

- Acceptation d'une connexion :

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

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

```
int accept(
```

```
    int sockfd,      /* socket de rendez-vous */
```

```
    struct sockaddr *addr, /* pointeur sur l'adresse de la socket connectée */
```

```
    socklen_t *addrlen /* pointeur sur la longueur de l'adresse */
```

```
);
```

- Demande de connexion :

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

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

```
int connect(
```

```
    int sockfd,      /* socket locale */
```

```
    const struct sockaddr *addr, /* pointeur sur l'adresse du serveur */
```

```
    socklen_t addrlen /* longueur de l'adresse */
```

```
);
```

4. Fonctions d'entrée/sortie sur un socket TCP

- Lecture :

```
#include <unistd.h>
```

```
ssize_t read(int fd, void *buf, size_t count);
```

- Ecriture :

```
#include <unistd.h>
```

```
ssize_t writeread(int fd, const void *buf, size_t count);
```

5. Manipulation des processus

- Récupération du numéro de processus du processus appelant :

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

```
#include <unistd.h>
```

```
pid_t getpid(void);
```