



QUI SUIS-JE ?

HAI404I
Support de TP
Manuel résumé de getaddrinfo

Getaddrinfo(....)

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

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

```
#include <netdb.h>
```

```
int getaddrinfo(const char *node, const char *service,  
               const struct addrinfo *hints,  
               struct addrinfo **res);
```

```
void freeaddrinfo(struct addrinfo *res);
```

```
const char *gai_strerror(int errcode);
```

Que fait `getaddrinfo()`

- Étant donnés *node* et *service*, qui identifient un *hôte Internet* et un *service*, **`getaddrinfo()`** renvoie une ou plusieurs structure *addrinfo*, chacune d'entre elles contenant une adresse Internet .

Node

- soit **une adresse réseau** en format numérique
 - décimal pointé pour l'IPv4, comme prise en charge par inet_aton(3) ;
 - hexadécimal pour l'IPv6, comme prise en charge par inet_pton(3),
- soit **un nom d'hôte**, dont l'adresse réseau est alors résolue.
- Si le membre *hints.ai_flags* contient l'attribut **AI_NUMERICHOST** alors *node* devra être une adresse réseau numérique.
- L'attribut **AI_NUMERICHOST** empêche toute tentative, éventuellement longue, de résolution de nom d'hôte.

Service

- définit le port dans chacune des structures d'adresses renvoyées.
- Si cet argument est un nom de service (consultez services(5)), il est convertit en son numéro de port correspondant.
- Cet argument peut également être indiqué sous forme décimale, qui est simplement converti en binaire.
- Si *service* est NULL, le numéro de port des adresses de socket renvoyées n'est pas initialisé.
- Si **AI_NUMERICSERV** est indiqué dans *hints.ai_flags* et si *service* n'est pas NULL, *service* doit pointer vers une chaîne contenant une valeur numérique de port.

Hints

- pointe sur une structure *addrinfo* qui spécifie les critères de sélection des structures d'adresses de sockets renvoyées dans la liste pointée par **res**.
- Si **hints** n'est pas NULL, il doit pointer sur une structure *addrinfo* dont les membres *ai_family*, *ai_socktype*, et *ai_protocol* indiquent les critères limitant l'ensemble d'adresses de sockets renvoyées par **getaddrinfo**
- Tous les autres membres de la structure pointée par **hints** doivent contenir 0 ou être des pointeurs NULL.
- Spécifier **hints à NULL** est équivalent à définir
 - *ai_socktype* et *ai_protocol* à 0,
 - *ai_family* à **AF_UNSPEC** et
 - *ai_flags* à **(AI_V4MAPPED | AI_ADDRCONFIG)**.

La fonction **getaddrinfo()**

- **alloue et initialise une liste chaînée de structures addrinfo,**
 - une pour chaque adresse réseau correspondant à *node* et *service*, soumise aux restrictions imposées par l'argument *hints*, et renvoie dans *res* un pointeur sur le début de la liste.
- Les éléments de la liste sont chaînés par le champ *ai_next*.
 - Il y a plusieurs raisons pour lesquelles la liste chaînée peut avoir plus d'une structure *addrinfo* : l'hôte réseau est « multihomed » ; le même service est accessible depuis plusieurs protocoles (par exemple **AF_INET** et **AF_INET6**) ou accessible depuis plusieurs types de socket (par exemple une adresse de type **SOCK_STREAM** et une autre de type **SOCK_DGRAM**).
- Normalement, l'application essaie d'utiliser les adresses dans l'ordre où elles sont renvoyées.
 - La fonction de tri utilisée dans **getaddrinfo()** est définie dans la RFC 3484 ; le tri peut être configuré pour un système particulier avec le fichier */etc/gai.conf* (disponible depuis la glibc 2.5).

getaddrinfo()

- **getaddrinfo()** renvoie
 - 0 si elle réussit,
 - Ou un des codes d'erreur non nuls

La structure addrinfo

```
struct addrinfo {  
    int         ai_flags;  
    int         ai_family;  
    int         ai_socktype;  
    int         ai_protocol;  
    socklen_t   ai_addrlen;  
    struct sockaddr *ai_addr;  
    char        *ai_canonname;  
    struct addrinfo *ai_next;  
};
```

Addrinfo et hints

- Si *hints* n'est pas NULL, il doit pointer sur une structure *addrinfo* dont les membres *ai_family*, *ai_socktype*, et *ai_protocol* indiquent les critères limitant l'ensemble d'adresses de sockets renvoyées par **getaddrinfo()**, de la façon suivante :
- *ai_family*
 - Ce champ indique la famille d'adresse désirée des adresses renvoyées.
 - Les valeurs valides de ce champ incluent **AF_INET** et **AF_INET6**. La valeur **AF_UNSPEC** indique que **getaddrinfo()** doit renvoyer les adresses de socket de n'importe quelle famille d'adresses (par exemple, IPv4 ou IPv6) pouvant être utilisées avec *node* et *service*.
- *ai_socktype*
 - Ce champ indique le type préféré de socket, par exemple **SOCK_STREAM** ou **SOCK_DGRAM**.
 - Mettre 0 dans ce champ indique que **getaddrinfo()** peut renvoyer n'importe quel type d'adresses de socket.

Addrinfo et hints

- *ai_Protocol*
 - Ce champ indique le protocole des adresses de socket renvoyées.
 - Mettre 0 dans ce champ indique que **getaddrinfo()** peut renvoyer des adresses de socket de n'importe quel type.
- *ai_flags*
 - Ce champ indique des options supplémentaires. Divers attributs sont regroupés par un OU binaire.
 - Tous les autres membres de la structure pointée par *hints* doivent contenir 0 ou être des pointeurs NULL.
- Spécifier *hints* à NULL est équivalent à définir *ai_socktype* et *ai_protocol* à 0, *ai_family* à **AF_UNSPEC** et *ai_flags* à **(AI_V4MAPPED | AI_ADDRCONFIG)**.