

AI_NUMERICSERV。参数 service 默认可以是服务名或端口号。这个标志强制参数 service 为端口号。

AI_PASSIVE。getaddrinfo 默认返回套接字地址，客户端可以在调用 connect 时用作主动套接字。这个标志告诉该函数，返回的套接字地址可能被服务器用作监听套接字。在这种情况下，参数 host 应该为 NULL。得到的套接字地址结构中的地址字段会是通配符地址(wildcard address)，告诉内核这个服务器会接受发送到该主机所有 IP 地址的请求。这是所有示例服务器所期望的行为。

```

struct addrinfo {
    int             ai_flags;      /* Hints argument flags */
    int             ai_family;     /* First arg to socket function */
    int             ai_socktype;   /* Second arg to socket function */
    int             ai_protocol;   /* Third arg to socket function */
    char            *ai_canonname; /* Canonical hostname */
    size_t          ai_addrlen;    /* Size of ai_addr struct */
    struct sockaddr *ai_addr;      /* Ptr to socket address structure */
    struct addrinfo *ai_next;      /* Ptr to next item in linked list */
};

```

code/netp/netpfragments.c

图 11-16 getaddrinfo 使用的 addrinfo 结构

当 getaddrinfo 创建输出列表中的 addrinfo 结构时，会填写每个字段，除了 ai_flags。ai_addr 字段指向一个套接字地址结构，ai_addrlen 字段给出这个套接字地址结构的大小，而 ai_next 字段指向列表中下一个 addrinfo 结构。其他字段描述这个套接字地址的各种属性。

getaddrinfo 一个很好的方面是 addrinfo 结构中的字段是不透明的，即它们可以直接传递给套接字接口中的函数，应用程序代码无需再做任何处理。例如，ai_family、ai_socktype 和 ai_protocol 可以直接传递给 socket。类似地，ai_addr 和 ai_addrlen 可以直接传递给 connect 和 bind。这个强大的属性使得我们编写的客户端和服务端能够独立于某个特殊版本的 IP 协议。

2. getnameinfo 函数

getnameinfo 函数和 getaddrinfo 是相反的，将一个套接字地址结构转换成相应的主机和服务名字符串。它是已弃用的 gethostbyaddr 和 getservbyport 函数的新的替代品，和以前的那些函数不同，它是可重入和与协议无关的。

```

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

int getnameinfo(const struct sockaddr *sa, socklen_t salen,
                char *host, size_t hostlen,
                char *service, size_t servlen, int flags);

```

返回：如果成功则为 0，如果错误则为非零的错误代码。

参数 sa 指向大小为 salen 字节的套接字地址结构，host 指向大小为 hostlen 字节的缓冲区，service 指向大小为 servlen 字节的缓冲区。getnameinfo 函数将套接字地址结构 sa 转换成对应的主机和服务名字符串，并将它们复制到 host 和 service 缓冲区。如果 getnam-