



图 11-15 getaddrinfo 返回的数据结构

在客户端调用了 `getaddrinfo` 之后，会遍历这个列表，依次尝试每个套接字地址，直到调用 `socket` 和 `connect` 成功，建立起连接。类似地，服务器会尝试遍历列表中的每个套接字地址，直到调用 `socket` 和 `bind` 成功，描述符会被绑定到一个合法的套接字地址。为了避免内存泄漏，应用程序必须在最后调用 `freeaddrinfo`，释放该链表。如果 `getaddrinfo` 返回非零的错误代码，应用程序可以调用 `gai_strerror`，将该代码转换成消息字符串。

`getaddrinfo` 的 `host` 参数可以是域名，也可以是数字地址（如点分十进制 IP 地址）。`service` 参数可以是服务名（如 `http`），也可以是十进制端口号。如果不想把主机名转换成地址，可以把 `host` 设置为 `NULL`。对 `service` 来说也是一样。但是必须指定两者中至少一个。

可选的参数 `hints` 是一个 `addrinfo` 结构（见图 11-16），它提供对 `getaddrinfo` 返回的套接字地址列表的更好的控制。如果要传递 `hints` 参数，只能设置下列字段：`ai_family`、`ai_socktype`、`ai_protocol` 和 `ai_flags` 字段。其他字段必须设置为 0（或 `NULL`）。实际中，我们用 `memset` 将整个结构清零，然后有选择地设置一些字段：

- `getaddrinfo` 默认可以返回 IPv4 和 IPv6 套接字地址。`ai_family` 设置为 `AF_INET` 会将列表限制为 IPv4 地址；设置为 `AF_INET6` 则限制为 IPv6 地址。
- 对于 `host` 关联的每个地址，`getaddrinfo` 函数默认最多返回三个 `addrinfo` 结构，每个的 `ai_socktype` 字段不同：一个是连接，一个是数据报（本书未讲述），一个是原始套接字（本书未讲述）。`ai_socktype` 设置为 `SOCK_STREAM` 将列表限制为对每个地址最多一个 `addrinfo` 结构，该结构的套接字地址可以作为连接的一个端点。这是所有示例程序所期望的行为。
- `ai_flags` 字段是一个位掩码，可以进一步修改默认行为。可以把各种值用 OR 组合起来得到该掩码。下面是一些我们认为有用的值：

AI_ADDRCONFIG。如果在使用连接，就推荐使用这个标志 [34]。它要求只有当本地主机被配置为 IPv4 时，`getaddrinfo` 返回 IPv4 地址。对 IPv6 也是类似。

AI_CANONNAME。`ai_canonname` 字段默认为 `NULL`。如果设置了该标志，就是告诉 `getaddrinfo` 将列表中第一个 `addrinfo` 结构的 `ai_canonname` 字段指向 `host` 的权威（官方）名字（见图 11-15）。