Lecture 9

ADVANCED NAME AND ADDRESS CONVERSIONS

Functions

- getaddrinfo()
- gai_strerror()
- freeaddrinfo()
- getnameinfo()

getaddrinfo()

- **Hides all of protocol dependencies** in library function, in which they belong
- Application deals only with socket address structures that are filled in by getaddrinfo()

- Syntax

```
#include        <netdb.h>
int getaddrinfo(const char *hostname, const char *service,
    const struct addrinfo *hints, struct addrinfo **result);
```

- Returns
  - 0 if ok
  - nonzero on error

getaddrinfo() - cont

- Function returns thro result ptr, a ptr to a linked list of addrinfo structures, defined in <netdb.h>

```
struct addrinfo
{
        int ai_flags;       //AI_PASSIVE, AI_CANONNAME
        int ai_family;      //AF_xxx
        int ai_socktype; //SOCK_xxx
        int ai_protocol;    //0 or IPPROTO_xxx
        size_t ai_addrlen;          //length of ai_addr
        char *ai_canonname;     //ptr to canonical name of host
        struct sockaddr *ai_addr; //ptr to socket addr struct
        struct addrinfo *ai_next;   //ptr to next struct in linked
                                                //list
}
```

getaddrinfo() - cont

- Hostname:- can be either a hostname or address string
- Service:- either service name or decimal port no
- Hints :- either a NULL or pointer to addrinfo struct that caller fills in with hints about type of info that caller wants returned

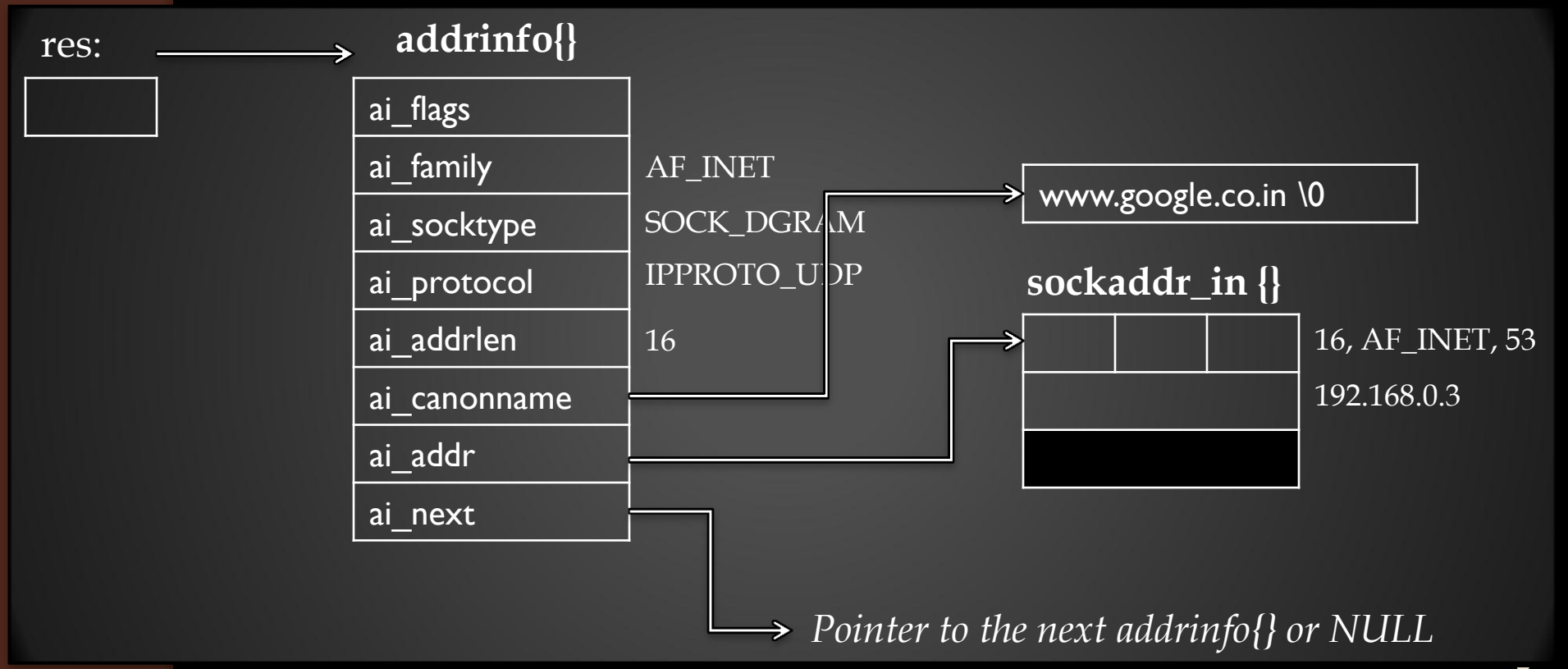
- Members of hints structure that can be set by caller are
  - ai_flags (AI_PASSIVE, AI_CANONNAME)
  - ai_family
  - ai_socktype
  - ai_protocol

getaddrinfo() - cont

- AI_PASSIVE :- socket will be used for passive open (listening)
- AI_CANONNAME:- tells function to return canonical (official) name of host
- If hints argument is NULL, function assumes a value of 0 for ai_flags, ai_socktype and ai_protocol, and value of AF_UNSPEC for ai_family

getaddrinfo() - cont

- Sample addrinfo{} structure – dynamically allocated by getaddrinfo()



gai_strerror()

- Takes one of the return values from getaddrinfo() as an argument and returns a pointer to corresponding error string

- Syntax

```
#include <netdb.h>
char* gai_strerror(int error);
```

freeaddrinfo()

- Storage returned by getaddrinfo(), addrinfo structures, ai_addr structures and ai_canonname are obtained dynamically by malloc()
- This storage is returned by calling freeaddrinfo()

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

- ai should point to first of the addrinfo structures returned by getaddrinfo()
- All structures in linked list are freed

Sample program

- Server program
  - dayserver_ind.c
- Client program
  - dayclient_ind.c
- Run in the same way as before…

- Kindly go through the contents in the next slide

getnameinfo() – 1/5

- Complement of getaddrinfo()
- It takes a socket address and returns a char string describing the host and another char string describing service
- Caller does not care about what type of protocol address is contained in socket address structure, as that detail is handled by the function

getnameinfo() – 2/5

```
#include <netdb.h>

int getnameinfo(const struct sockaddr *sockaddr,
socklen_t addrlen, char *host, size_t hostlen, char *serv,
size_t servlen, int   flags);
```

- Returns
  - 0 if ok
  - -1 on error

getnameinfo() – 3/5

- sockaddr:- points to socket address structure containing protocol addr to be converted to human readable string
- addrlen:- length of this structure
- If caller does not want host string returned, hostlen of 0 is specified
- The same is for the service

getnameinfo() – 4/5

- Flags values
  - NI_DGRAM:- datagram service. Specified when caller is dealing with datagram
  - NI_NAMEREQD:- cause error to be returned if hostname cannot be resolved. Can be used by servers that require client's IP address be mapped with hostname
  - NI_NOFQDN:- cause returned hostname to be truncated at first period(.)

getnameinfo() – 5/5

- NI_NUMERICHOST:- return numeric string for hostname
- NI_NUMERICSERV:- return numeric string for service

- Logical OR of multiple flags can be specified if they make sense together
- Eg. NI_DGRAM and NI_NUMERICHOST