NAME

res_parse, res_copy, res_print, res_free − parse and manipulate DNS resolver responses

SYNOPSIS

#include rr.h

res_response ∗ **res_parse(msg)**
char ∗msg;

res_response ∗ **res_copy(resp)**
res_response ∗resp;

res_print(resp)
res_response ∗resp;

res_free(resp)
res_response ∗resp;

DESCRIPTION

These routines are used for parsing the response from a call to the DNS *resolver* routines into a C structure suitable for programmer use, and to manipulate such structures.

Res_parse takes the result of a call to res_query or res_search. This is then parsed into an appropriate structure, and domain names unpacked into standard null-terminated C strings. It then returns a pointer to this structure.

Res_copy takes a pointer to a **res_response** structure, makes a copy of it and returns a pointer to the copy.

Res_print prints out the specified **res_response** structure, using a similar style to *nslookup*.

Res_free takes a pointer to a **res_response** structure, frees all memory allocated to it and clears the counts of the number of records of each type present.

DATA STRUCTURES

The structures used are as follows:

res_response {
HEADER **header;**
The header as defined in <**arpa/nameser.h**>.
s_question ∗∗**question;**
Pointer to an array of question records.
s_rr ∗∗**answer;**
Pointer to an array of answer resource records.
s_rr ∗∗**authority;**
Pointer to an array of nameserver resource records.
s_rr ∗∗**additional;**
Pointer to an array of additional resource records (the addresses of the authoritative nameservers).
}

s_question {
char ∗**qname;**
The name that the query was sent for.

u_short qtype
u_short qclass
The type and class of query, as defined in <**arpa/nameser.h**>.
}

s_rr {
char ***name;**
The name of the resource record.
u_short type
u_short class
The type and class of the record, as defined in <**arpa/nameser.h**>.
u_long ttl
How long (in seconds) this record may be cached.
u_long dlen;
The length of the type-specific data.
This need not be the same as the size of the *unpacked* data - particularly for domain names.
union u_rdata rdata;
The type-specific data (see **rr.h** for details).
}

RETURN VALUES

Res_parse and **res_copy** will return a pointer to the new structure, or a NULL pointer in case of errors. **Res_print** and **res_free** do not return anything.

WARNINGS

Res_parse now returns a pointer to a dynamically assigned area, (previous releases used a static area which which would be overwritten by subsequent calls).
The pointer returned by **res_parse** should be freed using **res_free** (contrary to the instructions with earlier releases of this library).

If **res_print** is called with a pointer that has been freed, it will (probably) print out an "empty" header. This is a Bad Thing to do anyway!

FILES

/usr/local/lib/libresparse.a

SEE ALSO

resolver(3), gethostbyname(3)
DNS and BIND

AUTHOR

Dave Shield <D.T.Shield@csc.liv.ac.uk>