

NAME

`pcap_next_ex`, `pcap_next` – read the next packet from a `pcap_t`

SYNOPSIS

```
#include <pcap/pcap.h>

int pcap_next_ex(pcap_t *p, struct pcap_pkthdr **pkt_header,
                 const u_char **pkt_data);
const u_char *pcap_next(pcap_t *p, struct pcap_pkthdr *h);
```

DESCRIPTION

pcap_next_ex() reads the next packet and returns a success/failure indication. If the packet was read without problems, the pointer pointed to by the *pkt_header* argument is set to point to the *pcap_pkthdr* struct for the packet, and the pointer pointed to by the *pkt_data* argument is set to point to the data in the packet. The *struct pcap_pkthdr* and the packet data are not to be freed by the caller, and are not guaranteed to be valid after the next call to **pcap_next_ex()**, **pcap_next()**, **pcap_loop(3PCAP)**, or **pcap_dispatch(3PCAP)**; if the code needs them to remain valid, it must make a copy of them.

pcap_next() reads the next packet (by calling **pcap_dispatch()** with a *cnt* of 1) and returns a *u_char* pointer to the data in that packet. The packet data is not to be freed by the caller, and is not guaranteed to be valid after the next call to **pcap_next_ex()**, **pcap_next()**, **pcap_loop()**, or **pcap_dispatch()**; if the code needs it to remain valid, it must make a copy of it. The *pcap_pkthdr* structure pointed to by *h* is filled in with the appropriate values for the packet.

The bytes of data from the packet begin with a link-layer header. The format of the link-layer header is indicated by the return value of the **pcap_datalink(3PCAP)** routine when handed the **pcap_t** value also passed to **pcap_loop()** or **pcap_dispatch()**. <https://www.tcpdump.org/linktypes.html> lists the values **pcap_datalink()** can return and describes the packet formats that correspond to those values. The value it returns will be valid for all packets received unless and until **pcap_set_datalink(3PCAP)** is called; after a successful call to **pcap_set_datalink()**, all subsequent packets will have a link-layer header of the type specified by the link-layer header type value passed to **pcap_set_datalink()**.

Do **NOT** assume that the packets for a given capture or “savefile” will have any given link-layer header type, such as **DLT_EN10MB** for Ethernet. For example, the “any” device on Linux will have a link-layer header type of **DLT_LINUX_SLL** or **DLT_LINUX_SLL2** even if all devices on the system at the time the “any” device is opened have some other data link type, such as **DLT_EN10MB** for Ethernet.

RETURN VALUE

pcap_next_ex() returns **1** if the packet was read without problems, **0** if packets are being read from a live capture and the packet buffer timeout expired, **PCAP_ERROR** if an error occurred while reading the packet, and **PCAP_ERROR_BREAK** if packets are being read from a “savefile” and there are no more packets to read from the savefile. If **PCAP_ERROR** is returned, **pcap_geterr(3PCAP)** or **pcap_peror(3PCAP)** may be called with *p* as an argument to fetch or display the error text.

pcap_next() returns a pointer to the packet data on success, and returns **NULL** if an error occurred, or if no packets were read from a live capture (if, for example, they were discarded because they didn’t pass the packet filter, or if, on platforms that support a packet buffer timeout that starts before any packets arrive, the timeout expires before any packets arrive, or if the file descriptor for the capture device is in non-blocking mode and no packets were available to be read), or if no more packets are available in a “savefile.” Unfortunately, there is no way to determine whether an error occurred or not.

SEE ALSO

pcap(3PCAP)