NAME

curl_multi_info_read - read multi stack informationals

SYNOPSIS

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

CURLMsg *curl_multi_info_read( CURLM *multi_handle,
                               int *msgs_in_queue);
```

DESCRIPTION

Ask the multi handle if there are any messages/informationals from the individual transfers. Messages may include informationals such as an error code from the transfer or just the fact that a transfer is completed. More details on these should be written down as well.

Repeated calls to this function will return a new struct each time, until a NULL is returned as a signal that there is no more to get at this point. The integer pointed to with *msgs_in_queue* will contain the number of remaining messages after this function was called.

The data the returned pointer points to will not survive calling curl_multi_cleanup().

The 'CURLMsg' struct is very simple and only contain very basic informations. If more involved information is wanted, the particular "easy handle" in present in that struct and can thus be used in subsequent regular curl_easy_getinfo() calls (or similar):

```
struct CURLMsg {
  CURLMSG msg;       /* what this message means */
  CURL *easy_handle; /* the handle it concerns */
  union {
    void *whatever;    /* message-specific data */
    CURLcode result;   /* return code for transfer */
  } data;
};
```

RETURN VALUE

A pointer to a filled-in struct, or NULL if it failed or ran out of structs. It also writes the number of messages left in the queue (after this read) in the integer the second argument points to.

SEE ALSO

curl_multi_cleanup(3), **curl_multi_init**(3), **curl_multi_perform**(3)