

sabato 24 dicembre 2022 10:24

voce 025

- è una variante della malloc che ti permette di allocare vettori di oggetti. CALLOC restituisce un vettore di oggetti AZZERATO!
LA MALLOC NO.

- ## REALLOC

chiama malloc, memcpy, una free, e restituire il nuovo indirizzo;
↳ del nuovo vettore;

VECTOR.C

VECTOR H

[illegible]

```

    free_vector(people);
    return 0;
}

```

VECTOR.H

```

#pragma once
#include <stdlib.h>
#include <stdbool.h>

struct array;

extern void *init_vector(size_t nmemb, size_t size);
extern bool insert_at(struct array *arr, size_t pos, void *el);
extern bool push_back(struct array *arr, void *el);
extern bool read_from(struct array *arr, size_t pos, void *buffer);
extern bool extract_from(struct array *arr, size_t pos, void *buffer);
extern void free_vector(struct array *arr);

```

```

    memcpy(from, el, arr->size);
    arr->count++;
    assert(arr->count <= arr->slots);
    return true;
}

bool read_from(struct array *arr, size_t pos, void *buffer)
{
    if(pos < arr->count) {
        memcpy(buffer, (char *)arr->arr + pos * arr->size, arr->size);
        return true;
    }
    return false;
}

bool extract_from(struct array *arr, size_t pos, void *buffer)
{
    char *from, *to;
    size_t size;
    if(!read_from(arr, pos, buffer))
        return false;
    if(pos + 1 < arr->slots) {
        from = (char *)arr->arr + (pos + 1) * arr->size;
        to = from - arr->size;
        size = (arr->count - pos) * arr->size;
        memmove(to, from, size);
    }
    arr->count--;
    if(arr->count < arr->slots / 2)
        resize(arr, arr->slots / 2);
    return true;
}

```

```

CFLAGS=-Wall -Wextra -g
TARGET=dvec
SRCS=main.c vector.c
OBJS=$(SRCS:.c=.o)

%.o:%.c
    $(CC) $(CFLAGS) $< -c -o $@

all: $(OBJS)
    $(CC) $(CFLAGS) $(OBJS) -o $(TARGET)

.PHONY:clean
clean:
    -$(RM) $(TARGET) $(OBJS)

```