

# Buddy allocator con bitmap

*Studente: Dimitrova Donika Todorova matricola: 1805163*

## What

Il progetto consiste nella realizzazione di un buddy allocator utilizzando la struttura ausiliaria bitmap. Con l'implementazione tramite bitmap si risparmia molto spazio per l'allocazione della struttura che gestisce la memoria.

## How

Attraverso la struttura della bitmap ho traccia dei blocchi liberi e occupati della memoria. Tale struttura rappresenta l'albero della memoria nel quale ogni bit corrisponde a un nodo e ogni livello dell'albero corrisponde a un livello del buddy allocator.

Ogni nodo può assumere al massimo 2 valori:

- 0 quando la zona di memoria non è disponibile
- 1 la zona di memoria associata è disponibile

Un nodo potrebbe non essere disponibile a seguito di una richiesta precedente, il nodo è stato diviso per creare due buddy a un livello inferiore oppure il nodo genitore deve essere diviso per creare il buddy al livello richiesto.

Ad ogni livello della struttura corrisponde una dimensione e tutti i nodi di quel livello corrispondono a blocchi di memoria tutti della stessa dimensione.

Le principali funzioni del buddy allocator che servono all'utente sono 2: malloc e free.

Malloc:

Dopo la richiesta di una certa dimensione, l'allocatore esegue alcuni passi:

- 1) Stabilisce il livello a cui si trova la memoria da allocare corrispondente alla dimensione di byte da allocare.
- 2) Se un nodo ha il valore del bit 1 non deve essere considerato, mentre se il bit è 0 prima di poterlo scegliere devo verificare che tutti i bit dei padri del nodo abbiano valore 0.
- 3) Una volta trovato il nodo adatto setto il bit a 1 e prima di restituire l'indirizzo di memoria associato, scrivo nei primi 8 byte dell'indirizzo da restituire l'indirizzo del buddy (ossia del nodo fratello, figlio dello stesso padre) e restituisco l'indirizzo spostato di 8 byte. Poiché devo sempre restituire l'indirizzo del buddy nei primi 8 byte, alla size richiesta nella malloc aggiungo sempre 8.

Free:

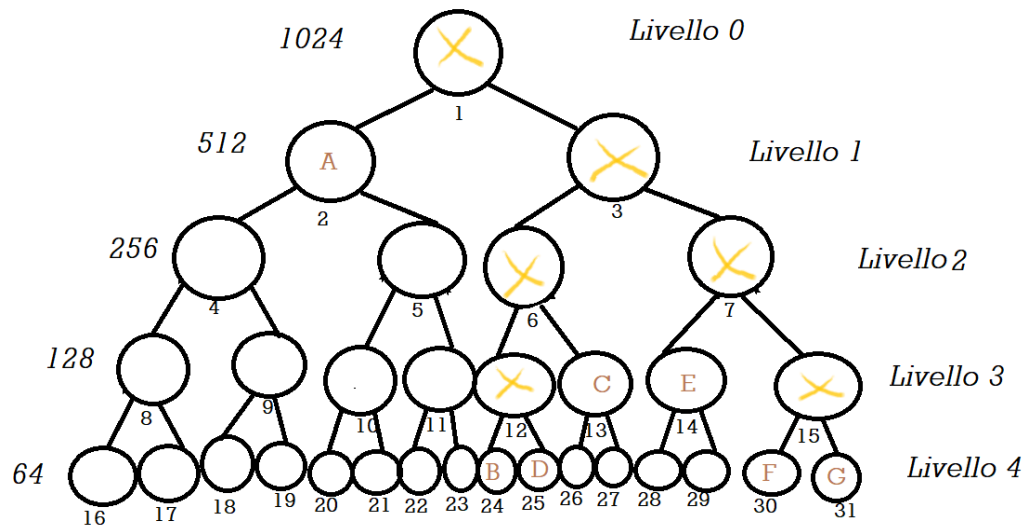
Dato l'indirizzo da liberare, devo trovare l'indice corrispondente nella bitmap e settarlo a zero. Per trovare l'indice nella bitmap devo usare l'indirizzo del buddy del blocco che voglio liberare scritto nei primi 8 byte del blocco da liberare.

Tramite l'indirizzo del buddy mi calcolo la dimensione del blocco che voglio liberare e da questa trovo l'indice corrispondente nella bitmap e lo setto a zero, e setto anche tutti i padri dell'indice zero.

## How to run

Si può compilare nel terminale attraverso l'istruzione *make*. Dopodiché si può eseguire uno dei 6 test forniti attraverso il comando: `./buddy_allocator_test_ [1-6]`.

## Esempio di Test



./buddy\_allocator\_test\_1

- A. 374
- B. 12
- C. 66
- D. 33
- E. 67
- F. 1
- G. 1

Il segno giallo indica che il nodo genitore è stato diviso per creare due buddy ed entrambi hanno il bit settato a 1.