

Zarządzanie pamięcią operacyjną: Buddy system

Autor: Paweł Mnich

Sposób przydzielania pamięci za pomocą buddy system

1. Gdy pamięć jest pusta, to traktowana jest jako jeden ciągły blok.
2. Wraz z rezerwowaniem bloku pamięci o rozmiarze m pamięć jest dzielona w połowie na bloki o rozmiarze 2^n do momentu aż zostanie znaleziony wolny blok gdzie $2^{(n-1)} < m \leq 2^n$ (np. dla $m=20B$ będzie to $16B < 20B \leq 32B$), blok ten jest wtedy oznaczany jako zarezerwowany.
3. Gdy zarezerwowana pamięć jest zwalniana to jeśli to możliwe to bloki pamięci o tym samym rozmiarze są łączone z powrotem w jeden blok.

Przykład

1. Jest pamięć o rozmiarze 64B.



2. Ktoś chce zarezerwować 12B. Ponieważ $64B > 2 * 12B$ pamięć jest dzielona na pół.



3. Ponieważ $32B > 2 * 12B$ jeden z bloków pamięci jest ponownie dzielony na pół.



4. Ponieważ $16B \leq 12B * 2$ to nie możemy podzielić bloku pamięci na pół, więc rezerwowany jest jeden z wolnych bloków:

16B (zarez.)	16B	32B
--------------	-----	-----

5. Ktoś teraz chce zarezerwować 30B. Ponieważ blok 64B został podzielony na 2 bloki 32B to schodzimy niżej i sprawdzamy czy są wolne. Pierwszy blok 32B został podzielony na 16B, a 16B to za mało dla zarezerwowania 30B, więc przechodzimy do drugiego bloku 32B. Blok ten jest wolny oraz $32B \leq 30B * 2$ czyli można go zarezerwować.

16B (zarez.)	16B	32B (zarez.)
--------------	-----	--------------

6. Ktoś kto zarezerwował na początku 12B skończył używać tej pamięci więc chce ją zwolnić:

16B	16B	32B (zarez.)
-----	-----	--------------

7. Są teraz dwa wolne bloki obok siebie 16B, więc są one łączone w jeden na wypadek gdyby chciano zarezerwować przestrzeń między 17B a 32B:



I zostaje wolny blok 32B oraz 32B zarezerwowany.

Implementacja

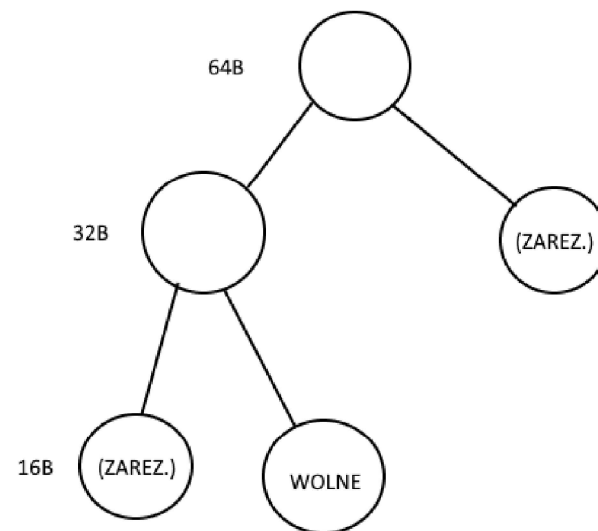
- Wartość określająca minimalny rozmiar bloku pamięci
- Wartość określająca rozmiar pamięci
- Tablica wartości Byte o długości rozmiaru pamięci
- Drzewo binarne określające jak pamięć jest podzielona, jakie jej części są wolne oraz jakie części są zajęte

Przykładowy wygląd drzewa binarnego

16B (zarez.)

16B

32B (zarez.)



Funkcje

- `int zarezerwuj_pamiec(int rozmiar) /* Rezerwuje pamięć o podanym rozmiarze i zwraca adres bajtu w którym się zaczyna */`
- `void zwolnij_pamiec(int adres) /* zwalnia blok pamięci zaczynający się pod danym adresem */`
- `void zapisz_bajt(Byte wartosc, int adres) /* zapisuje wartość bajta na podany adres */`
- `Byte odczytaj_bajt(int adres) /* odczytuje wartość bajta na podanym adresie */`

Funkcje

- `void pokaz_pamiec()` – wypisuje zawartość całej pamięci do konsoli
- `void pokaz_podzial()` – wypisuje podział pamięci, części wolne, części zajęte i ich rozmiar

Wywoływanie funkcji

Funkcje wymienione do rezerwacji, zwalniania, odczytywania i zapisywania pamięci są publiczne. Wszystkie te funkcje wywoływane są przez moduł zarządzania pamięcią wirtualną, zależnie od potrzeb.

Funkcje do pokazywania stanu pamięci będą publiczne i wywoływane przez moduł interfejsu.