

Wstęp do programowania w języku C

Grupa MSz w poniedziałki

Lista 7 na zajęcia 2.12.2019

Zadanie 1. (10 punktów na pierwszej pracowni, 5 punktów na drugiej)

Zdefiniuj makro `PRINTBIN`, które generuje kod wypisujący znaki zakodowane przez podaną w parametrze sekwencję wartości zapisanych binarnie. Każdy bajt jest opisany przez kolejne 8 znaków (zer i jedynek), od najbardziej do najmniej znaczącego bitu. Jeśli znaków nie dzieli się przez 8, to przyjmujemy, że początek jest dopełniony zerami. Makro powinno udawać pojedynczą instrukcję, co najlepiej zrobić opakowując całość w `do {...} while(0)`. Ponadto kod generowany przez makro nie powinien wywoływać żadnych funkcji poza bibliotecznymi.

Przykład wykorzystania:

```
PRINTBIN(11000010110001001100011); // abc
```

Wskazówka: Podany parametr makra możemy zamienić na ciąg znakowy poprzez dopisanie na początku znaku `#`.

```
#define PRINT(napis) \
    printf("%s\n",#napis);
```

Wtedy `PRINT(tekst);` jest równoważne `printf("%s\n","tekst");`.

Test:

```
for (int i = 0; i < 10; i++)
    if (i%2)
        PRINTBIN(111010001011010010100100001010); else
        PRINTBIN(1010);
```

Zadanie 2. (10 punktów)

Napisz program, który określi maksymalną wielkość bloku pamięci, który można zarezerwować w danym momencie przy pomocy `malloc`. Należy zastosować wyszukiwanie binarne oraz użyć typu `size_t` do przechowywania rozmiaru bloku.

Wskazówka: `malloc` rezerwuje blok pamięci podanego rozmiaru w bajtach. Jeśli zwraca `NULL` to znaczy, że się nie udało; w przeciwnym wypadku należy przekazać otrzymany wskaźnik do funkcji `free`, żeby zwolnić pamięć.

Uwaga: To czy rezerwacja bloku pamięci się udaje może być niedeterministyczne i kończyć się różnie nawet w trakcie tego samego uruchomienia programu. Tym nie trzeba się przejmować, a wynik traktujemy orientacyjnie.