

# Systemy operacyjne 2016

## Lista zadań nr 7

### Na zajęcia ?

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Stallings (wydanie siódme): 7.1 - 7.4, 7A, 8.1
- Tanenbaum (wydanie czwarte): 3.1 - 3.3

**UWAGA!** W trakcie prezentacji rozwiązań należy zdefiniować i wyjaśnić pojęcia, które zostały oznaczone **wytluszczoną** czcionką.

**Zadanie 1.** Podaj przykład występowania zjawiska **fragmentacji zewnętrznej** i **fragmentacji wewnętrznej**. Jakie problemy związane z **ciągłym przydziałem** pamięci fizycznej rozwiązuje stronicowanie? Do czego służy **kompaktowanie**?

**Zadanie 2.** Jak już wiemy, funkcje `malloc` i `free` nie są wywołaniami systemowymi. Jakie szczególne właściwości mają **bloki** przydzielane funkcjami biblioteki standardowej? Czy tak przydzielona pamięć może podlegać kompaktowaniu? Wyjaśnij kiedy występuje błąd **wycieku pamięci** i opisz pozostałe błędy związane z używaniem dynamicznego przydziału pamięci.

**Zadanie 3.** Systemy operacyjne udostępniają wywołania systemowe do przydziału ciągów stron na użytek bibliotecznego algorytmu zarządzania pamięcią. Opisz jak wykorzystać w tym celu wywołania `sbrk(2)` oraz `mmap(2)`. Czemu z tego drugiego korzysta się częściej?

**Zadanie 4.** Opisz listowy algorytm zarządzania pamięcią w przydziale ciągłym z polityką **first-fit**. W jakim celu wykorzystuje się technikę **złączania** wolnych bloków? Odpowiedz na pytania:

- jak wygląda struktura danych przechowująca listę wolnych i zajętych bloków?
- jak przebiegają operacje `malloc` i `free`?
- jaka jest oczekiwana złożoność czasowa operacji?
- jaki jest narzut pamięciowy struktur danych?
- jaki jest maksymalny rozmiar **nieużytków**?

**UWAGA!** Algorytm zarządzania pamięcią może korzystać tylko i wyłącznie z uprzednio przydzielonego jednego ciągłego obszaru pamięci – również do przechowywania dynamicznych struktur danych. Jakiegokolwiek mechanizmu przydziału pamięci poza tym obszarem są niedozwolone!

**Zadanie 5.** Wyjaśnij różnice między politykami **first-fit**, **next-fit**, **best-fit**, **worst-fit** stosowanymi w listowym algorytmie przydziału pamięci. W jakich warunkach poszczególne polityki mogą zapobiegać fragmentacji? Rozważ **czas życia** i rozmiar bloków. Zauważ, że listę wolnych bloków możemy również przeszukiwać od końca.

**Zadanie 6.** Opisz algorytm przydziału pamięci korzystający z **bitmapy** i odpowiedz na pytania z zadania 3. Jak przyspieszyć wyszukiwanie wolnych bloków? Załóż, że należy minimalizować ilość potencjalnych chybień w pamięć podręczną.

**Zadanie 7.** Jądro systemu operacyjnego implementuje specjalny algorytm do przydziału stron pamięci fizycznej. Najczęściej jest to wariant **systemu bliźniaków** (ang. *buddy systems*). Wyjaśnij jego działanie i odpowiedz na pytania z zadania 3. Czemu do takich zastosowań wykorzystuje się ten algorytm zamiast ogólnych metod przydziału?

**Zadanie 8.** Bazując na strategii **quick-fit** zaproponuj własną implementację algorytmu przydziału pamięci. Możesz wykorzystać dodatkowe struktury danych – np. zbalansowane drzewa binarne. Czy gorliwe złączanie wolnych bloków jest zawsze dobrym pomysłem? Wyjaśnij działanie swojego algorytmu i odpowiedz na pytania z zadania 3.

**Zadanie 9 (bonus).** Jak **algorytm płytowy**<sup>1</sup> (ang. *slab allocator*) rozwija ideę algorytmu przydziału bazującego na bitmapie? W jakich warunkach ten rodzaj algorytmu jest szczególnie przydatny?

**Zadanie 10 (bonus).** Dlaczego algorytmy przydziału pamięci często cierpią na zjawisko **rywalizacji o blokady** (ang. *lock contention*)? Opisz pobieżnie struktury danych wykorzystywane przez bibliotekę **thread-caching malloc**<sup>2</sup>. Wyjaśnij techniki używane do efektywnej obsługi żądań przydziału bloków w aplikacjach wielowątkowych.

---

<sup>1</sup><http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.4759>

<sup>2</sup><http://goog-perftools.sourceforge.net/doc/tcmalloc.html>