

S0, pracownia, wybór zadań nieobowiązkowych

Wyciąg z materiałów wykładowych, czyli **wybór zadań**, których wykonaniu – oprócz zadania obowiązkowego – można poświęcić czas i uwagę na pracowni w ramach pracy własnej (**dotyczy na pewno grup zpl**).

1. Ile jest Linuxów w 320 dystrybucjach systemu Linux? (Artykuł).
2. Zainstaluj w swoim komputerze 3, 4 lub więcej systemów operacyjnych (w tym jeden z Microsoftu). Zadbaj o wygodny sposób ich rozruchu. UWAGA! Nie używaj maszyn wirtualnych. Pamiętaj: wiesz, co robisz.
3. Wyszukaj w Sieci najnowszą wersję firmowego BIOS-u Twojego komputera i zastąp nim stary BIOS w swoim komputerze. Pamiętaj: wiesz, co robisz.
4. Zbuduj sobie w wolnej chwili i oprogramuj system komputerowy. Wskazówka. Ponieważ jeden ze starszych "Windowsów" (NT 3.0) liczył 40000 plików źródłowych w języku C (5,6 mln wierszy kodu), więc żeby przyspieszyć robotę zajrzyj do książki Nisana i Schockena *Elementy systemów komputerowych* (WNT, Warszawa 2008). Jeśli mimo to nie starczy Ci czasu na zrobienie całego komputera, to możesz się ograniczyć do jakiegoś jego fragmentu – wtedy bez oprogramowania.
5. Napisz moduł obsługi wybranego urządzenia zewnętrznego. Może być małe i nietypowe.
6. Przejrzyj *The Central File Extensions Registry* (<http://www.fileinfo.com>), wybierz kilka typów plików, zapoznaj się z ich przeznaczeniem, sprowadź z Sieci (wiesz, co robisz) kilka przykładów takich plików i programy interpretujące ich formaty. Poeksperymentuj. Opisz wyniki eksperymentów. Dla szczególnie ambitnych: napisz program konwersji między dwoma wybranymi formatami plików.
7. Zapoznaj się z dorobkiem S.P. Jobsa (1955-2011). Prześledź opinie jego entuzjastów i przeciwników. Wyrób sobie własny pogląd o tym menadżerze i informatyku. Przedstaw to w artykule.
8. Wyucz się języka interpretera poleceń w użytkowanym przez siebie systemie operacyjnym. Udowodnij, że to się opłaca, pisząc i wdrażając jeden lub kilka niebanalnych skryptów systemowych.
9. Znajdź gdzieś starego peceta z systemem MS-DOS. Napisz dla niego program rezydentny.
10. Przeczytaj dla przyjemności artykuły autorów (lub ich dotyczące), do których kierują odsyłacze na tej stronie (np. Dijkstra, Hopper, Kernighan, Thompson, Ritchie i in.).
11. Jak w zadaniu 10, tylko w odniesieniu do oprogramowania, np. Linux, Mach, POSIX, Solaris, Unix, Windows).
12. Popraw błędne definicje maszyny wirtualnej i środowiska uruchomieniowego (w drugim przypadku chodzi o *runtime system*, czyli o system wykonawczy. (Będziesz wielk{a|i}, jeśli przekonasz wikipedystów, że masz rację, że jest inaczej niż w *Samych swoich*).
13. Przeczytaj *Bajkę o złotej rybce* Aleksandra Puszkina (w przekładzie Juliana Tuwima). Wyciągnij wnioski.
14. Napisz krótką rozprawę, lecz o takiej sile oddziaływania (jak to osiągniesz?), że ogół przestanie mówić i pisać "aplikacja webowa" (łebska aplikacja?) i wróci do określeń "aplikacja oparta na usłudze WWW, czyli aplikacja WWW". Jeśli Ci się to uda, zaliczysz S0 przed terminem.
15. Wyszukaj w *Silberschatzu* i zaprogramuj wszystkie zadania, w których sugeruje się użycie wątków.
16. Napisz uniwersalny symulator algorytmów planowania procesów.

17. Odpowiedz, kim jest piąty filozof, którego wizerunek przedstawiono na wykładzie. Odpowiedź ma być oczywiście dobrze uargumentowana. Przeczytaj do poduszki *Powiastrki filozoficzne* Woltera w przekładzie Tadeusza Boya-Żeleńskiego. Czy problem: "Co oni jedzą?" jest do przezwyciężenia? (A może jest NP-zupełny? :)).

18. Znajdź w prawie równie paradoksalny przepis (lub układ przepisów) jak w słynnej regulacji obowiązującej swego czasu w stanie Kansas.

19. Wyjaśnij algorytm strusia.

20. Przeczytaj wiersz *Rycerz Krzykański* Juliana Tuwima. Wyciągnij wnioski.

21. Napisz (i przetestuj) własny podsystem nakładkowania.

22. Oprogramuj podsystem zarządzanie pamięcią zorganizowaną w n stosów. Początkowo stosy są puste. W trakcie działania systemu stosy przyrastają (lub maleją) w różnym tempie. W odniesieniu do danego stosu zamówienia mogą być różnej wielkości. Jeśli wyrażone w bajtach zamówienie na pamięć dla któregoś ze stosów nie może być zrealizowane, ponieważ blok wolnej pamięci powyżej stosu jest za mały, należy wykonać procedurę reorganizacji całej pamięci, kierując się **zasadą przydziałów proporcjonalnych**. To znaczy, że po reorganizacji stos największy otrzyma powyżej swojego obszaru najwięcej wolnej pamięci, a stos najmniejszy – najmniej; oczywiście po zaspokojeniu żądania, które doprowadziło do kompleksowej reorganizacji pamięci. Jeśli okaże się, że zamówienie jest niewykonalne, podsystem zarządzania pamięcią zgłasza błąd krytyczny. Aby wszechstronnie przetestować podsystem, zaprojektuj i zrealizuj odpowiednie symulacje.

23. Zainstaluj w swoim komputerze RAM-dysk. Docień zalety, wskaż wady.

24. Znajdź dzienniki prowadzone w Twoim systemie operacyjnym i obejrzyj zawarte w nich wpisy. Zinterpretuj ich znaczenie.

25. Zadanie zaproponowane i wykonane przez **Macieja Przybeckiego**. Zbuduj dystrybucję Linuxa, wychodząc od dostępnego w Sieci kodu źródłowego tego systemu, tzn. utwórz niezbędne struktury dyskowe, skompiluj, skonsoliduj i przetestuj wszystkie niezbędne pliki systemu i sporządź stosowny raport. Zmierz, ile czasu zajmie Ci wykonanie tego zadania.

Ciąg dalszy [być może] nastąpi.