Lista 5 (Lab) Termin wysłania na SVN do 10.12.2017

1. *(10pt)* Napisz w języku C własną implementacje funkcji **printf** i **scanf** (nazwijmy je **myprintf** i **myscanf**). Funkcje nie mogą wykorzystywać, żadnych funkcji bibliotecznych (**atoi**, **fprintf**, **fscanf** itp.) oraz makr **va_start**, **va_arg** i **va_end** (np. możesz skorzystać z wyjaśnienia tutaj) oraz mogą wykorzystać wywołania systemowe **read** i **write** z odpowiednim standardowym deskryptorem. Program **należy** skompilować na maszynę 32-bitową tzn. **gcc -m32** np. dla 64-bitowego systemu ArchLinux trzeba zainstalować pakiet gcc-multilib z repozytorium multilib. W funkcjach wystarczy zaimplementować "%s", "%d", "%x" i "%b", gdzie w naszej implementacji "%s" wyświetla ciąg znaków, "%d" liczbę w systemie dziesiętnym, "%x" liczbę w systemie szesnastkowych oraz "%b" liczbę w systemie binarnym.
2. *(10pt)* Napisz w języku C wielowątkową wersję **mnożenia macierzy boolowskich**. Program powinien pobierać z linii komend wielkość macierzy (wypełniać ją losowymi wartościami 0 lub 1, patrz **man 3 random**) oraz liczbę wątków, która powinna zostać uruchomiona do mnożenia. Zaimplementuj program tak, że każdy wątek pracuje na osobnym wierszu, jeśli jeden skończy pracę to dalej pracuje na następnym wolnym wierszu oraz pamiętaj, że pojedynczy iloczyn skalarny (wiersz razy kolumna) może zostać ustalony wcześniej nawet po pierwszej koniunkcji. Pamiętaj, że przy dostępie do zmiennych współdzielonych mogą wystąpić wyścigi!
3. *(10pt)* Napisz program w języku C podobny do **minitalk** bez wykorzystywania dodatkowych procesów (**fork**) lub wątków. Wykorzystaj wywołanie systemowe **select** oraz **gniazda**. Program ma być w tym zadaniu prostym komunikatorem internetowym. Na początku następuje proste logowanie (bez haseł) do serwera, wtedy wyświetlone są loginy wszystkich zalogowanych użytkowników. Następnie możemy wysłać wiadomość do dowolnego użytkownika i tylko do niego.
4. *(8pt)* Załóż system plików FAT (podobnie jak na wykładzie). Utwórz na nim kilka plików (co najmniej jeden większy od wielkości pojedynczego klastra) i pokaż jak są przechowywane w systemie plików (**hexdump -C**) np. jak zmieniła się tablica alokacji (FAT), tablica katalogów (root directory entry) oraz gdzie znajduje się zawartość plików. Usuń wybrany plik i pokaż jak zmienił się system plików. Co należy zrobić aby odzyskać skasowany plik?
5. *(15pt)** Napisz w języku C program, wykorzystujący FUSE, który pozwala szyfrować dane na dysku. Dokładnie uproszczoną wersję programu EncFS. Podobnie jak program **encfs**tworzymy dwa katalogi np. **/tmp/crypt-raw**, który przechowywuje szyfrowane dane oraz **/tmp/crypt** który przedstawia dane z katalogu **/tmp/crypt-raw** w postaci odkodowanej. Do kodowania należy wykorzystać algorytm AES lub inny aktualnie uważany za bezpieczny.