

# Kurs administrowania systemem Linux 2019

Lista zadań na pracownię nr 14

Na zajęcia 3 czerwca 2019

**Zadanie 1 (Na Dzień Dziecka, 3 pkt).** Zainstaluj emulator systemu DOS, np. popularny DOSBox (DOSBOX.COM) lub (nieco zapomniany) DOSEMU (DOSEMU.ORG) albo prawdziwy system DOS, np. FreeDOS (freedos.org), na emulatorze trybu rzeczywistego procesora x86, np. na Qemu, VirtualBoksie lub PCEmu albo na rzeczywistej maszynie (można zrobić *dual boot* dla Linuksa i FreeDOS, np. za pomocą memdisk). Zainstaluj w nim podstawowe narzędzia, w tym „the masterpiece of assembler programming” — The Volkov Commander (vvv.kiev.ua). Ze znanych serwisów oferujących klasyczne gry (DOSGAMES.COM, classicdosgames.com) wybierz kilka, zainstaluj je i uruchom. Przygotuj krótką prezentację możliwości systemu DOS i jego emulatorów. Przejrzyj jakiś dobry podręcznik programowania dla systemu DOS (np. Ray Duncan, *Advanced MS-DOS Programming: The Microsoft Guide for Assembly Language and C Programmers*, Microsoft Press 1988) i przygotuj porównanie systemu DOS i systemów uniksowych, w szczególności opisz zadania systemu DOS i odpowiedz na pytania:

- Czy DOS zarządza pamięcią?
- Czy pamięć jest zwirtualizowana?
- Czy jest w DOS-ie ochrona pamięci?
- Czym jest proces w DOS-ie?
- Czy DOS dokonuje podziału czasu pomiędzy procesami?
- Jak odbywa się zarządzanie procesami?
- Jak jest zorganizowana obsługa wywołań systemowych?
- Jak architektura procesora 8088 wpłynęła na strukturę systemu DOS?

**Zadanie 2 (1 pkt).** Przeczytaj dokumentację pakietu *GNU Diffutils* i przygotuj jego zwięzłe omówienie. Wybierz jakikolwiek krótki program swojego autorstwa. Wprowadź w nim niewielką modyfikację. Wygeneruj za pomocą programu `diff` odpowiednią łatę. Oryginalną wersję programu zmodyfikuj w innym miejscu. Za pomocą programu `patch` nałóż na tak zmodyfikowany tekst wcześniejszą poprawkę. Przygotuj się do zademonstrowania takiego procesu nakładania łat podczas zajęć.

**Zadanie 3 (4 pkt).** Z oficjalnego repozytorium `ftp.gnu.org` projektu GNU skopiuj archiwum najnowszej wersji plików źródłowych programu *GNU Hello*. Pamiętaj o sprawdzeniu podpisu cyfrowego pakietu! Przygotuj krótkie omówienie sposobu weryfikacji skopiowanego oprogramowania. Jak sprawdzić wiarygodność breloka z kluczem publicznym GPG oferowanego do pobrania z repozytorium GNU?

Użyj standardowej metody kompilowania i instalowania oprogramowania i zainstaluj program *GNU Hello* na swoim komputerze.

Za pomocą programu `checkinstall(1)` wygeneruj pakiet zawierający program *GNU Hello*.

Przeczytaj dokumentację programu *GNU Hello* oraz dokumenty *GNU Coding Standards* oraz *Information for Maintainers of GNU Software* i przygotuj omówienie zasad tworzenia oprogramowania w ramach projektu GNU.

Zmodyfikuj następnie program *GNU Hello* dodając opcję `-c, --color`, która powoduje, że program na początku i końcu wypisywanego tekstu dodaje sekwencje sterujące ANSI zmieniające kolor wypisywanego tekstu. Pamiętaj także o właściwym zmodyfikowaniu dokumentacji! Wygeneruj następnie łatę zawierającą tę modyfikację.

**Zadanie 4 (4 pkt).** Zapoznaj się z narzędziami Autotools (Autoconf, Automake i Libtool) i przygotuj omówienie sposobu konfigurowania projektu za ich pomocą. Użyj pakietu *GNU Hello* jako przykładu.

**Zadanie 5 (4 pkt).** W tym zadaniu wcielisz się w rolę *package maintainera* dystrybucji bazującej na Debianie. Utwórz pakiet źródłowy zawierający program *GNU Hello*. Dodaj do niego łąkę z zadania 3. Wygeneruj pakiet binarny. Przygotuj omówienie sposobu zarządzania pakietami oraz korzystania z narzędzi `apt-build`, `dch`, `dh_make` i `debuild` w Debianie.