Zapoznajemy się z podstawowymi komendami w linuxie poznajemy środowisko oraz sposoby kompilacji pierwszych programów. Laboratorium to ma na celu przypomnienie sobie podstawy działania w środowisku systemu operacyjnego Linux.

Podstawy Linuxa

1. Logowanie
   Po uruchomieniu komputerów wybrać system operacyjny operacyjny LINUX, najnowszą wersję. Zalogować się loginem i hasłem podanym przez prowadzącego.
2. Katalog domowy - tu zaczynamy

- pwd - gdzie jesteśmy
- ls -la - wypisanie zawartości bieżącego katalogu
- grep {ciąg} {maska plików} - wyszukiwanie ciągu znaków wśród plików zadanych określonych przez maskę.
- zmienne środowiskowe
- echo $NASZA_ZMIENNA - wyświetli wartość zmiennej NASZA_ZMIENNA
- alias ll="ls -la" - tworzenie skrótów, tu przykład utworzenia skrótu ll wykonującego polecenie ls -la.
- env i set - operacje na zmiennych środowiskowych
- export $ZMIENNA - wyeksportowanie zmiennej środowiskowej.
- hostname - wyświetlenie nazwy komputera
- id - wyświetlenie identyfikatorów użytkownika.
- logname - wyświetlenie loginu użytkownika.

3. .bashrc i .profile - pliki konfiguracyjne
   W katalogu domowym użytkownika znaleźć można wiele plików konfiguracyjnych. Dwa z nich .bashrec oraz .profile (w uproszczeniu) przetwarzane są one podczas otwierania nowego terminala lub logowania.
   Proste ćwiczenie:

- Wpisać do pliku konfiguracyjnego .bashrc następującą linię:
  export BLABLA="ala ma kota"
- Otworzyć nowy terminal.
- Sprawdzić zawartość zmiennej BLABLA za pomocą:
  echo $BLABLA

4. Poruszanie się

- cd <ścieżka> - przejście do podanego katalogu
- mv <nazwa1><nazwa2> - zmiana nazwy/położenia pliku
- cp <nazwa1><nazwa2> - kopiowanie pliku
- rm <plik> - usuwanie pliku. Opcja -rf pozwoli usunąć też cały katalog.
- mkdir <katalog> - tworzenie katalogu
- rmdir <katalog> - usuwanie katalogu
  Stworzyć w katalogu domowym katalog A oraz B. W katalogu A katalog AA oraz BB. W katalogu AA katalog AAA oraz BBB. W katalogu B katalog CC. Teraz wykonać cd (bez parametrów, znajdziemy się w katalogu domowym) i jednym poleceniem przejść do katalogu BBB. Sprawdzić gdzie jesteśmy pwd. Następnie jednym poleceniem przejść do katalogu CC. Usunąć katalog CC. Usunąć katalog A. Usunąć katalog B.

5. Pliki

- cat <plik> - wyświetl zawartość pliku
- head <plik> - wyświetl pierwsze kilka linii pliku
- tail <plik> - wyświetl ostatnie kilka linii pliku. Przydatna opcja -f przy śledzeniu plików logowania.
- ln -s <plik> <link> - tworzenie linku symbolicznego do pliku.
- more <plik> - wyświetlanie zawartości pliku z zatrzymywaniem po każdym ekranie.
- wc <plik> - zliczanie znaków, słów i linii w pliku.

6. Procesy

- ps - pokazanie bieżących procesów w systemie, często używane w takim zestawieniu ps -efa |grep student - pokaż wszystkie procesy systemu w formie długiej ale przepuść przez filtr tylko te które mają nazwę "student"

- kill, killall - głównie używane do usuwania procesów. Choć kill ogólnie służy do wysyłania sygnałów.
- top - pokazanie statystyk procesów najbardziej "zasobożernych"
- znaczenie znaków & > >> | - uruchomienie w tle i przekierowania
- fg, bg - operacje na procesach w tle.
  Uruchomić polecenie ls > plik.txt co będzie zawartością pliku plik.txt? Można posłużyć się poleceniem more lub cat. Jaka jest wielkość pliku?
  Teraz wykonać to samo ale z przekierowaniem >>. Jaka jest wielkość pliku? Powtórzyć operacje z pojedyńczym przekierowaniem i jeszcze raz sprawdzić wielkość pliku.

7. Inne

- tty - wyświetlenie bieżącego urządzenia terminala
- type <program> - pełna ścieżka do programu
- uname -a - informacje na temat komputera łącznie z numerem jądra.

8. Edytor Vim

- zaczynamy pisanie:a,i,A,I,o,O
- kończymy pisanie:esc
- poruszamy się:strzałki, h,j,k,l,pgup,pgdwn,n,N,``,ma ~a
- kasujemy:x,d,D
- kopiujemy: y
- wklejamy: p,P
- szukamy: /,n,N,*,#
- zamieniamy: cw,r
- wychodzimy: q,q!,wq,q!
- ustawienia: :set np :set nu :syntax on
- plik konfiguracyjny: .vimrc
- Więcej informacji można znaleźć przygotowanym przezemnie mini podręczniku do VIM

9. Montowanie nośników
   W celu zamontowania dyskietki wykonujemy polecenie
   mount /mnt/floppy
   Od tego momentu w katalogu /mnt/floppy będzie dostępna zawartość dyskietki
   Przed wyjęciem dyskietki należy ją odmontować:
   umount /mnt/floppy
   Pendrive jeżeli nie zamontuje się automatycznie to trzeba go zamontować i odmontować w podobny sposób.
   W razie problemów trzeba pytać prowadzącego zajęcia.
   W niektórych systemach montowanie odbywa się w katalogu /media zamiast /mnt
10. Poznajemy kompilator gcc
    Napisać prosty program typu "hello world" i go skompilować. Kompilacja odbywa się za pomocą kompilatora gcc.

    gcc plik.c -o program

    Jeżeli kompilacja wymaga podania ścieżek poszukiwań innych niż domyślne lub dołączenia jakiejś biblioteki to można zastosować opcje:

- -I<sciezka_includow>
- -L<sciezka_bibliotek>
- -l<biblioteka>

Jeżeli chcemy by kompilator powiadamiał nas o wszystkich błędach i ostrzeżeniach należy włączyć opcję:

- -Wall

11. Przekazywanie parametrów do programu - lista argumentów (0,5pkt)
    Lista argumentów składa się z tablicy wskaźników na napisy.

```
main (argc, argv)
     int argc; /* liczba napisów */
     char *argv[]; /* wskaźnik do tablicy napisów zapis **argv jest równoznaczny */
     {}
```

Napisać program wyświetlający parametry wejściowe. Każdy parametr powinien być wyświetlony w

osobnym wierszu i w odwrotnej kolejności podawania. Np.:

```
./program1 ala ma kota a kot ma ale
        ale
        ma
        kot
        a
        kota
        ma
        ala
        program1
```

Uwaga! Zerowy parametr to nazwa programu.

12. Lista zmiennych środowiska (0,5pkt)

- zawsze gdy wykonywany jest program przekazywana jest też lista zmiennych środowiskowych
- umieszczana jest w przestrzeni danych procesu.
- jest to tablica wskaźników do napisów w języku C
- zakończona jest wskaźnikiem NULL.
- napis ma zazwyczaj postać: zmienna=napis
- do programu w C można przekazać ją jako trzeci parametr w main()
- funkcja inicjująca w C tworzy też zmienną zewnętrzną
  extern char **environ ;
- wskazuje na to samo co envp
- w rzeczywistości najłatwiej zrobić to przez
  char *getenv(char *nazwa_zmiennej);
- poniższy program ma przekazaną listę zmiennych przez trzeci argument funkcji main oraz ją wyświetla:

```
main(argc,argv,envp)
    int argc;
    char *argv[];
    char *envp[];
    {
    int ii;
    for (ii=0; envp[ii] != (char *) 0; ii++)
        printf("%s\n",envp[ii]);
    exit(0);
    }
```

- Należy przepisać i uruchomić

13. Numery identyfikacyjne tzw. Pid'y (0,5pkt)
    Jest to unikalny numer przydzielany przez system każdemu procesowi. Możemy go uzyskać za pomocą funkcji:

    getpid();

    Każdy proces posiada także swój proces macierzysty. Pid procesu macierzystego można uzyskać za pomocą funkcji

    getppid();

    Napisać programik wypisujący identyfikator procesu oraz identyfikator procesu macierzystego.
    Sprawdzić do czego służy funkcja getpgrp() i setpgrp().

14. Identyfikatory użytkownika i grupy (0,5pkt)
    Można je pobrać za pomocą getuid() oraz getgid().
    Napisać program (lub dopisać do poprzedniego) wyświetlający identyfikatory użytkownika i grupy.
    Sprawdzić do czego są funkcje geteuid() i getegid()

15. Pliki (1,5pkt)

- każdy plik ma wiele atrybutów
- do odczytania atrybutów pliku służą stat i fstat

```
#include <sys/types.h>
#include <sys/stat.h>
int stat(char *pathname, struct stat * buf); /* po nazwię */
int fstat(int *fildes, struct stat *buf); /* po deskryptorze */
struct stat {
    ushort st_mode; /* typ pliku, prawa dostępu do pliku */
    ino_t st_ino; /* numer i-węzła */
```

```
    dev_t st_dev; /* identyfikator urządzenia, na którym jest pozycja w katalogu
odpowiadająca temu plikowi */
    short st_nlink; /*liczba dowiązań */
    ushort st_uid; /* identyfikator użytkownika */
    ushort st_gid; /* identyfikator grupy */
    dev_t st_rdev; /*identyfikator urządzenia dla znakowych lub blokowych plików
specjalnych */
    off_t st_size; /* rozmiar pliku w bajtach */
    time_t st_atime; /*czas ostatniego dostępu do pliku */
    time_t st_mtime; /*czas ostatniej modyfikacji pliku */
    tme_t st_ctime; /* czas ostatniej zmiany stanu pliku */
    long st_blksize; /* optymalny rozmiar bloku dla operacji na plikach; tylko 4.3BSD */
    long st_block; /*bieżąca liczba przydzielonych bloków; tylko w 4.3 BSD */
    };
```

Napisać program który wyświetli wszystkie możliwe dane na temat pliku którego nazwę przekażemy jako parametr. Sprawdzić czy podana wyżej struktura odpowiada tej zwracanej w systemie Linux.
Zmodyfikować program by wyświetlał informacje na temat wszystkich podanych plików jako parametry.

16. Sprawozdanie.
    Sporzadzić sprawozdanie w formie podręcznej " ściągawki" z powyższych poleceń. Sprawdzić jakie można dodawać opcje do różnych poleceń. Wyszukać inne polecenia i opisać.

    Kilka uwag!
    Sprawozdania proszę przysyłać mailem najpóźniej do końca dnia następnego. Po tym czasie sprawozdania nie będą oceniane.
    Sprawozdanie może być w postaci dokumentu txt lub OpenOffice
    Sprawozdanie powinno zawierać datę, numer i temat laboratorium którego dotyczy, grupę oraz imiona i nazwiska zespołu.
    Maile należy opatrzyć tematem z numerem grupy i numerem laboratorim a także by były podpisane, anonimów też nie będę otwierał. Ogólnie proszę się dostosować do tego.
    Do sprawozdań powinny być dołączone kody programów z ewentualnym opisem kompilacji.
    Treść sprawozdań może być różna w zależności od rodzaju ćwiczeń. Podam jednak kilka uniwersalnych wskazówek co zwykle powinno się znaleść w sprawozdaniu i co może być oceniane:

- Wspomniane wyżej dane.
- Przebieg ćwiczenia, wraz ze zrzutami (w formie txt) ekranu, odpowiedziami na zadane pytania, przebieg testów jakie przeprowadziliśmy.
- Wnioski. Czego się nauczyliśmy, co się udało a czego się nie udało wykonać, z czym były największe problemy.
- Kody programów.