

XXXIII OI, zawody II stopnia – Krótkie wprowadzenie do Linuksa

Wstęp

Drodzy zawodnicy!

Mamy świadomość, że II etap Olimpiady dla części z Was może być jednym z pierwszych kontaktów z Linuksem. Z myślą o tych osobach powstał ten krótki podręcznik, który mamy nadzieję pomoże Wam odnaleźć się w tym środowisku.

Ponadto na II etapie korzystacie z komputerów, które nie zostały skonfigurowane przez Was, przez co mogą wystąpić różne sytuacje, w których komputer nie działa tak, jak byście się tego spodziewali. W tym podręczniku przedstawimy więc także szereg problemów, na które możecie się natknąć, i sposobów, jak je rozwiązać, aby zaoszczędzić czas i uniknąć niepotrzebnego stresu.

Pisanie programów

Do pisania swojego kodu możesz wykorzystać jedno z dostępnych środowisk programistycznych takich jak Geany czy VS Codium. Aby uruchomić dowolne z nich wybierz z menu Programy podmenu Programowanie.

Do tworzenia swoich rozwiązań możesz też wykorzystać proste edytory tekstowe takie jak Gedit czy Kate. Znajdziesz je w menu Programy, w podmenu Akcesoria.

Terminal

W środowisku Linux ważnym narzędziem jest terminal. Aby go uruchomić, z menu Programy wybierz Emulator terminala albo po prostu kliknij odpowiednią ikonkę na pasku.

Po uruchomieniu będziesz w swoim folderze domowym czyli `/home/zawodnik`.

- Aby przejść do podfolderu wpisz:

```
$ cd nazwa_podfolderu
```

- Aby wrócić do folderu nadrzędnego wpisz:

```
$ cd ..
```

- W każdym momencie możesz sprawdzić w jakim folderze się znajdujesz:

```
$ pwd
```

- Żeby zobaczyć instrukcję obsługi jakiegoś programu, możesz wykorzystać program `man`. Na przykład:

```
$ man g++
```

- Strzałkami w górę i w dół możesz przechodzić po historii wykonanych poleceń.

Kompilacja w C++

Aby skompilować swój program napisany w języku C++, w terminalu wejdź do folderu, w którym zapisałeś swoje rozwiązanie i uruchom poniższą komendę:

```
$ g++ -O3 -static -o nazwapliku nazwapliku.cpp -std=c++23
```

XXXIII OI, zawody II stopnia – Krótkie wprowadzenie do Linuksa

W wyniku jej działania, w folderze, w którym się znajdujesz, powstanie plik o nazwie `nazwapliku`.

Warto rozważyć dodanie flagi kompilatora `-Wall`. Jej użycie spowoduje bardziej szczegółowe wypisywanie ostrzeżeń podczas kompilacji co pozwala odnaleźć wiele błędów.

Jeśli swój kod piszesz w środowisku programistycznym takim jak Geany czy Code::Blocks, to do kompilacji możesz użyć wbudowanej w niego opcji (np. w Geany wystarczy wybrać z menu Zbuduj opcję Zbuduj).

Uruchamianie programów w C++

Aby uruchomić swoje rozwiązanie uruchom terminal oraz wejdź do folderu, w którym zapisałeś swój kod. Jeśli tego wcześniej nie zrobiłeś, skompiluj rozwiązanie. Wpisz:

- `$ ./nazwaprogramu`
aby uruchomić swój program
- `$ ./nazwaprogramu < plik_testowy.in`
aby uruchomić swój program oraz przekierować zawartość pliku `plik_testowy.in` na standardowe wejście
- `$ ./nazwaprogramu < plik_testowy.in > wynik.out`
aby uruchomić swój program, przekierować zawartość pliku `plik_testowy.in` na standardowe wejście oraz wynik działania programu zapisać do pliku `wynik.out`
- `$ time ./nazwaprogramu < plik_testowy.in > wynik.out`
aby uruchomić swój program, przekierować zawartość pliku `plik_testowy.in` na standardowe wejście, wynik działania programu zapisać do pliku `wynik.out` oraz dodatkowo wyświetlić czas działania programu.
- `$ oiejq ./nazwaprogramu < plik_testowy.in > wynik.out`
aby uruchomić swój program w wirtualnym środowisku emulującym działanie procesora (w podobnym środowisku Jury będzie oceniać rozwiązania).

Uruchamianie programów w Pythonie

Wszystko działa tak jak powyżej, tylko zamiast `./nazwaprogramu` należy użyć

```
python3 nazwaprogramu.py
```

Testowanie

Aby przetestować swój program na testach przykładowych możesz użyć skryptu `./ocen`. W tym celu:

- zapisz swoje rozwiązanie w folderze `/home/zawodnik/rozw` z nazwą odpowiadającą trzyliterowemu kodowi zadania (np. `abc`) i rozszerzeniem odpowiadającym językowi, w którym piszesz
- w terminalu wejdź do ww. folderu
- uruchom komendę

```
$ ./ocen abc
```

kompilującą i uruchamiającą rozwiązanie wybranego zadania na testach przykładowych.

Problemy z terminalem

Jeśli chcesz zaznaczyć i skopiować pewien tekst z terminala, aby wkleić go gdzie indziej, możesz użyć w tym celu skrótu klawiszowego **Ctrl**+**Shift**+**C**. Podobnie, do wklejenia do terminala skopiowanego wcześniej tekstu możesz użyć kombinacji **Ctrl**+**Shift**+**V**. W szczególności warto użyć tego triku do kompilacji, żeby nie przepisywać komendy z pliku PDF z ustaleniami technicznymi. Co bardzo ważne, użycie w terminalu skrótu **Ctrl**+**C** daje zupełnie inny efekt niż skopiowanie zaznaczonego tekstu (jest to opisane w jednym z następnych akapitów).

UWAGA: W przypadku niektórych wersji terminala (np. uruchamianej przez Code::Blocks po wybraniu opcji „build and run”) ten sposób może nie działać. W takim przypadku spróbuj wkleić dane za pomocą wciśnięcia (nie przekręcenia – dosłownie wciśnięcia) kółka myszy albo za pomocą kombinacji **Shift**+**Insert**.

Jeżeli terminal w ogóle nie wygląda, jakby przyjmował znaki, to bardzo możliwe, że został zamrożony. Zamrożenie terminala najprawdopodobniej nastąpiło poprzez wprowadzenie skrótu klawiszowego **Ctrl**+**S** w oknie terminala. Aby odmrozić tak zamrożony terminal, możesz użyć skrótu **Ctrl**+**Q**. Zwróć uwagę na to, że odmrożenie terminala wyśle do niego wszystkie naciśnięcia klawiszy, które nastąpiły w trakcie zamrożenia.

Możesz znaleźć się w sytuacji, w której uruchomiony program (np. skompilowany kod Twojego rozwiązania zadania) działa bardzo długo (np. w wyniku nieskończonej pętli) i chcesz go awaryjnie zakończyć. Możesz to uczynić poprzez naciśnięcie w terminalu **Ctrl**+**C**. Ta kombinacja powinna zakończyć aktualnie działający w terminalu program.

Może tak się zdarzyć, że uruchomiony program oczekuje na więcej wejścia, a my nie mamy ochoty go już dostarczać. W takiej sytuacji wypada poinformować program, że już więcej od nas nie dostanie. Robi się to skrótem klawiszowym **Ctrl**+**D**. Czasami, jeżeli ostatnim wprowadzonym znakiem nie była nowa linia (klawiszem **Enter**), to należy nacisnąć **Ctrl**+**D** dwukrotnie. Należy uważać, żeby nie nacisnąć go za wiele razy, bo wtedy koniec pliku zostanie wysłany również powłoce i może zamknąć terminal.

Może się również zdarzyć, że podczas próby kompilacji rozwiązania otrzymasz komunikat

```
g++: error: unrecognized command-line option '-O3'
```

pomimo tego, że flaga `-O3` jest zawarta w oficjalnych wytycznych dotyczących kompilacji. Upewnij się wówczas, że podana przez Ciebie flaga to „wielkie O trzy”, a nie „zero trzy”.

Ogólne problemy

Jeżeli występuje efekt „ekran porusza się wraz z myszką”, to najpewniej przypadkowo włączyłeś lupę. Można ją wyłączyć poprzez jednoczesne przytrzymanie klawisza **Alt** oraz kręcenie w dół kółkiem myszy.

Czasem może się zdarzyć, że ekran komputera staje się czarny i wyświetla na górze niewielkie napisy podobne do „login zawodnik”. Najprawdopodobniej stało się to w wyniku naciśnięcia kombinacji klawiszy **Ctrl**+**Alt**+**Fk**, gdzie *k* jest pewną liczbą (przykładowo zostało wciśnięte **Ctrl**+**Alt**+**F3**). Aby przywrócić powłokę graficzną, wciśnij **Ctrl**+**Alt**+**F7**.