

Programowanie Systemowe i Współbieżne

Laboratorium 1: Pliki

Zadanie 1

Napisz program, który stworzy plik podany jako pierwszy argument wywołania z linii poleceń, a następnie zapisze do niego wartość drugiego argumentu.

Zadanie 2

Napisz program, który otworzy istniejący plik tekstowy w systemie i zamieni wszystkie wystąpienia litery "x" na "X".

Zadanie 3

Napisz program kopiujący zawartość pliku o nazwie podanej jako pierwszy parametr do pliku którego nazwa podana jest jako drugi parametr.

Zadanie 4

Napisz program, który wypisze zawartość pliku podanego jako argument od tył (jak program tac).

Zadanie 5

Napisz program wyszukiujący najdłuższą linię w pliku i podający ilość znaków w tej linii.

Zadanie 5

Napisz program określający rozmiar pliku (kilku plików o nazwach podanych jako parametry wejściowy)

Zadanie 6

Napisz program wypisujący od końca zawartość pliku

- a) znakami
- b) liniami

Zadanie 7

Napisz procedurę kopiowania ostatnich 10 znaków, słów i ostatnich 10 linii jednego pliku do innego.

Zadanie 8

Napisz program do rozpoznawania czy plik o podanej nazwie jest plikiem tekstowym (plik tekstowy zawiera znaki o kodach 0-127 – można w tym celu użyć funkcji isascii)

Zadanie 9

Napisz program konwertujący małe litery na duże w pliku o nazwie podanej jako parametr. Jeśli nazwa pliku nie została podana, dane powinny zostać odczytane ze standardowego wejścia.

Zadanie 10

Napisz program, który w pliku o nazwie podanej jako ostatni argument zapisze połączoną zawartość wszystkich plików których nazwy zostały podane w linii poleceń przed ostatnim argumentem.

Zadanie 11

Napisz program liczący ile jest słów w pliku (zakładamy, że słowa składają się z małych i dużych liter alfabetu oraz cyfr i znaku podkreślenia, a wszystkie pozostałe znaki są separatorami słów). Jeśli nazwa pliku nie została podana, dane powinny zostać odczytane ze standardowego wejścia.

Zadanie 12

Napisz program do wyznaczania częstości występowania liter w pliku tekstowym o nazwie podanej jako argument linii poleceń. Wynikiem działania programu powinien być wydruk na standardowym wyjściu określający procentową zawartość poszczególnych liter w całym tekście z pominięciem białych znaków oraz znaków interpunkcji.

Zadanie 13

Napisz program do filtrowania linii tekstu odczytywanego ze standardowego wejścia w taki sposób, że jeśli linia odczytanego tekstu zawiera łańcuch znaków przekazanych jako argument linii poleceń, to jest ona zapisywana na standardowym wyjściu, w przeciwnym przypadku jest pomijana.

Zadanie 14

Napisz program zmieniający kolejność znaków w każdej linii pliku o nazwie podanej jako parametr.

Zadanie 15

Napisz program do znajdowania łańcucha znaków podanego jako pierwszy argument linii poleceń w plikach o nazwach podanych jako pozostałe argumenty linii poleceń. Program powinien informować o nazwie pliku i miejscu (numer linii, numer znaku w linii), w którym dany łańcuch został znaleziony.

Zadanie 16

Napisz program do formatowania akapitów tekstu w plikach o nazwach podanych jako argumenty linii poleceń w taki sposób, żeby długość linii (liczba znaków w linii) nie przekraczała wartości podanej jako pierwszy argument linii poleceń. Przyjmując, że w wyniku formatowania nie może nastąpić podział słowa pomiędzy dwa wiersze i że separatorem formatowanego akapitu jest pusta linia.

Zadanie 17

Napisz program do porównywania plików o nazwach przekazanych jako argumenty. Wynikiem działania programu ma być komunikat że pliki są identyczne, pliki różnią się od znaku nr<nr znaku> w linii<nr znaku linii> lub – gdy gdy jeden z plików zawiera treść drugiego uzupełnioną o jakieś dodatkowe znaki – plik <nazwa> zawiera<liczba>znaków więcej niż zawartość pliku <nazwa>

Zadanie 18

Napisz program wyświetlający zawartość katalogu, którego nazwa została podana jako argument linii poleceń (można skorzystać z funkcji getcwd i scandir)

Zadanie 19

Napisz program wyświetlający informację o pliku, którego nazwa została podana jako argument linii poleceń (odpowiadający poleceniu ls -l (można skorzystać z funkcji fstat, getpwuid, getgrgid, ctime).