

## Zadanie 1: Plik w formacie binarnym

Napisać program generujący plik wielkości  $64^4 \cdot 2^3 \cdot 1024^2$  bajtów, którego zawartością ma być ciąg liczb `double`. Liczby mają być losowo wygenerowane, i każda ma być „zwyczajnie poprawna” (`FP_NORMAL`).

Nazwa pliku wynikowego jest pierwszym argumentem pozycyjnym programu. Do generowania liczb losowych użyć pliku specjalnego `/dev/urandom`, a do weryfikacji liczb funkcji `fpclassify(3)`. Należy ograniczyć ilość operacji wejścia/wyjścia, korzystając z własnej tablicy. (Wypróbować różne rozmiary.)

## Zadanie 2: *Sparse ffles* (pliki rozrzedzone?)

Cel: zespół programów, potrafiących obsługiwać pliki z niezaalokowanymi blokami.

### 1. `puste_pole`

Program generuje plik o zadanej nazwie i rozmiarze, który zajmuje 0B w systemie plików, nie licząc *i-węzła*. (Poprawność działania można zweryfikować za pomocą programu `du`.) Jeżeli już istnieje plik o takiej nazwie, to program zgłasza błąd.

program:            `puste_pole`  
parametry:  
    `-r <rozmiar>`    wielkość pliku liczona w jednostkach 8 KB  
    `<ścieżka>`        lokalizacja pliku wynikowego

### 2. `zasiew`

Program rozmieszcza wartości, otrzymane za pomocą parametrów pozycyjnych, w losowych miejscach zadanego pliku. Pilnuje przy tym, by nie trafiły one na te same pozycje. Jeżeli nie jest to możliwe do spełnienia, tj. ilość wartości do rozmieszczenia przekracza rozmiar pliku, to program ma zgłosić błąd.

Program przyjmuje dowolną ilość parametrów pozycyjnych, których interpretacja zależy od parametru `-f`. Może on przyjąć jedną z dwóch wartości: `liczba` albo `tekst`. W pierwszym przypadku, każdy z parametrów pozycyjnych jest traktowany jako liczba o rozmiarze jednego bajta. Liczby mogą być podane w formatach odpowiadających podstawie 8, 10 lub 16 (odpowiednie prefiksy).

W przypadku `-f tekst`, każdy z parametrów pozycyjnych jest traktowany jako łańcuch tekstowy. Jako wartości do rozmieszczenia brane są kody ASCII wszystkich znaków, ze wszystkich parametrów.

program:            `zasiew`

parametry:

-s	<ścieżka>	lokalizacja pliku do modyfikacji,
-f	<format>	sposób interpretowania parametrów pozycyjnych, możliwe wartości: tekst lub liczba
val_1 ... val_N		dowolna ilość parametrów pozycyjnych.

### 3. plon

Program wyświetla zawartość pliku, sposób zoptymalizowany pod kątem plików składających się z długich ciągów jednakowych wartości.

Zadany plik jest przeglądany w celu lokalizacji bloków jednakowych wartości (ciąg jednoelementowy też jest blokiem). Kolejne znaleziska są opisywane liniami w formacie <wartość> <krotność wystąpienia>.

Przy czym, dla bajtów odpowiadających znakom drukowalnym, <wartość> jest wyświetlana jako ten znak, a dla pozostałych, jako liczba dziesiętna.

Specjalnym przypadkiem są niezaalokowane obszary pliku („dziury”). Gdy podany jest parametr -!, to są one traktowane, jak w zwyczajnych plikach (wypełnione zerami). W przeciwnym wypadku mają być opisywane jako <NULL> <wielkość obszaru>.

program: plon

parametry:

-!	(opcjonalny) zmiana interpretacji niezaalokowanych bloków; jeżeli jest podany, to zawartość pliku jest odczytywana tradycyjnie, w przeciwnym wypadku program informuje o zlokalizowanych „dziurach”,
<ścieżka>	lokalizacja badanego pliku.

### 4. rozrost

Program powiększa zaalokowane obszary pliku, poprzez dopisywanie z jednego lub obu końców tych samych wartości, które występują w danym bloku. Kierunek i rozmiar powiększania jest opisywany parametrem, który jest postaci <kierunek> : <rozmiar>.

*kierunek* może przyjąć jedną z trzech wartości: ante, post albo ambo. Oznaczają one przyrost przed, za lub z obu stron bloku.

Natomiast *rozmiar* jest liczbą całkowitą z dodaną opcjonalnie jednostką:

- B - bajt (wartość domyślna),
- bb - blok = 512B,
- K - kilobajt = 1024B.

W przypadku, gdy rozrost jest dwustronny i nowe obszary zachodzą na siebie, należy zadbać o to, by część wspólna była równo między nie rozdzielona.

program: rozrost

parametry:

-s <ścieżka> lokalizacja pliku do modyfikacji,

<ekspansja> [<ekspansja>] sposób rozrostu obszaru,

gdzie:

<ekspansja> jest postaci <kierunek> : <rozmiar>

Drugi parametr <ekspansja> jest dozwolony tylko w przypadku, gdy ma nastąpić rozrost w obu kierunkach (ante i post), ale z różnymi rozmiarami.