0: LINUX. Żadnego Minixa, zadanie realizujemy w Linuksie.  
  
1. Koncepcja powinna jednoznacznie definiować format pliku, ze szczególnym  uwzględnieniem struktury katalogu i ew. innych struktur danych. Mile  widziane ujęcie w tabeli kolejnych pól, z rozmiarami, przeznaczeniem, itp.  
  
2. Narzędzia umożliwiające poszczególne operacje (kopiowanie w obie  strony, tworzenie pliku, itd.) powinny być programami wykonywanymi z  konsoli. Nie akceptuję rozwiązania z "głównym programem zarządczym", czy  to mającym własną linię poleceń, czy menu, czy GUI... Proszę wzorować się  na klasycznych poleceniach linuksowych w stylu cp, mv, mkfs, itd. KAŻDY z  tych programów powinien poprawnie obsługiwać opcję "--help": jej  nieaktualność lub niekompletność będzie powodem do obniżenia oceny.  
  
3. Nie ma potrzeby obsługiwania nazw Unicode. Wystarczy wsparcie dla  zwykłego ASCII (choć oczywiście pójście w tę stronę nie jest zakazane ani  źle widziane - po prostu nie daje dodatkowych punktów). Niemniej nazwy powinny móc zawierać spację.  
  
4. To nie czasy MS-DOS. Można przewidzieć limit długości nazwy, ale nie  może on być zbyt mały (np. 256 znaków jest ok, 16 - nie). Efektywne  rozwiązania pozwalające nie rezerwować niepotrzebnie dużo miejsca na  długie nazwy, kiedy w praktyce większość jest krótka, są mile widziane.  
  
5. Polecenia kopiowania powinny pozwalać na kopiowanie jednym poleceniem  wielu plików, tak aby można było przy kopiowaniu na dysk skorzystać z  rozwijania "wildcardów" przez shell. W drugą stronę, gdzie shell nie  pomoże, obsługa wildcardów nie musi być implementowana, ale nadal trzeba  umożliwiać skopiowanie wielu plików za jednym zamachem.  
  
6. Wymagana właściwa obsługa argumentów. Plik o nazwie zaczynającej się od  "-" jest jak najbardziej legalny, musi być zapewniona możliwość jego  obsłużenia! Podpowiedź - w linuksie zwykle opcja "--" kończy obsługę  opcji, po niej wszystko jest brane dosłownie. Zwracam uwagę na getopt - to  dobre narzędzie do tego. Podobnie nazwy plików zawierające znaki specjalne  jak '\*' czy '?'. Utrudniają korzystanie z nich, bo - żeby shell ich nie  rozwijał - trzeba zapisywać takie nazwy w apostrofach... ale powinny móc  być zapisane na systemie plików i z niego odczytane.  
  
7. Pół punktu w ocenie implementacji jest zarezerwowane na ocenę właściwej  abstrakcji formatu pliku - podstawowe operacje powinny być częścią  biblioteki używanej przez poszczególne programy-polecenia. Implementacja w  której każdy program bezpośrednio manipuluje plikiem nie może liczyć na  więcej, niż 3,5 punktu.  
  
8. Głównym ograniczeniem jest pojemność dysku, nie katalogu. Owszem,  pojemność katalogu może być ograniczona, ale bez przesady. Jeśli stosujemy  bloki, to liczba bloków przy danej pojemności ogranicza liczbę możliwych  do utworzenia plików - pojemność katalogu musi być wystarczająca, musi  wystarczyć nawet, jeśli każdy plik zajmie tylko jeden blok. Jeśli nie  stosujemy bloków, katalog musi pomieścić wystarczająco wiele plików, żeby  dysk dało się całkowicie zapełnić plikami o rozmiarze ok. 4KB.  
  
9. Należy zademonstrować poprawne działanie systemu plików w co najmniej  dwóch przypadkach: małego, o pojemności ok. 100KB, oraz dużego, o  pojemności co najmniej 50MB. Niedopuszczalna jest odrębna obsługa obu  przypadków - struktura dysku i narzędzia powinny być odpowiednio  elastyczne. Fragmentację prezentujemy tylko na małym systemie, na dużym  wymagałoby to zbyt wielu operacji.  
  
10. System powinien być "przyzwoicie" efektywny. Miejsce na katalog może  oczywiście być rezerwowane "na sztywno", ale w takim przypadku jego  rozmiar powinien zależeć od rozmiaru dysku - katalog zdolny do obsłużenia  50MB rozdąłby system o pojemności 50KB do nieprzyzwoitych rozmiarów.  Dynamiczne rozszerzanie katalogu pozwala tego uniknąć, ale jest  zdecydowanie trudniejsze...  
  
11. (uff, ostatnie...) W ramach testów warto pokazać, czym różni się  system, na którym dokonano wielu różnych operacji (chociażby w celu  pokazania fragmentacji) od drugiego, o identycznym rozmiarze, na który  wgrano końcową zawartość tego pierwszego. Zależnie od koncepcji, struktury  systemu plików, itd. efekty mogą być bardzo różne.