

001.txt

X10

W hierarchicznym systemie plikow wystepuje wiele katalogow

TAK

NIE

002.txt

X01

W hierarchicznym systemie plikow wystepuje wiele katalogow glownych

TAK

NIE

003.txt

X01

W hierarchicznym systemie plikow uzytkownikom przypisane sa priorytety dostepu do plikow

TAK

NIE

004.txt

X01

W hierarchicznym systemie plikow uzytkownicy maja dostep tylko do swoich plikow

TAK

NIE

005.txt

X10

W hierarchicznym systemie plikow okresla sie sciezki dostepu do plikow

TAK

NIE

006.txt

X10

W hierarchicznym systemie plikow wystepuje jeden katalog glowny

TAK

NIE

007.txt

X10

W hierarchicznym systemie plikow okreslona jest dokladnie jedna sciezka bezwzgledna do pliku

TAK

NIE

008.txt

X10

W hierarchicznym systemie plikow wystepuja sciezki wzgledne i bezwzgledne dostepu do plikow

TAK
NIE

009.txt
X10
W hierarchicznym systemie plików mogą występować wielokrotne odwołania twarde do pliku
TAK
NIE

010.txt
X01
Bloki alokacji (klastry) w systemie plików dobierane są tak, aby zminimalizować fragmentację zewnętrzną
TAK
NIE

011.txt
X10
Bloki alokacji (klastry) w systemie plików dobierane są tak, aby zminimalizować fragmentację wewnętrzną
TAK
NIE

012.txt
X10
Bloki alokacji (klastry) w systemie plików są stałej wielkości w całym systemie plików
TAK
NIE

013.txt
X01
Bloki alokacji (klastry) w systemie plików są zawsze wielkości dwóch rekordów fizycznych (sektorów) na dysku
TAK
NIE

014.txt
X01
Bloki alokacji (klastry) w systemie plików powinny być jak największe, aby było ich w systemie niewiele
TAK
NIE

015.txt
X10

Bloki alokacji (klastry) w systemie plików numerowane są liczbami binarnymi stałej długości, co ogranicza ich liczbę

TAK

NIE

016.txt

X10

Sekwencyjny dostęp do pliku polega na czytaniu lub zapisie pliku rekord po rekordzie w stałej kolejności

TAK

NIE

017.txt

X01

Sekwencyjny dostęp do pliku pozwala na występowanie plików nieciągłych, z niezaalokowanymi dziurami

TAK

NIE

018.txt

X01

Sekwencyjny dostęp do pliku jest możliwy tylko w plikach z rekordami stałej długości

TAK

NIE

019.txt

X01

Sekwencyjny dostęp do pliku nie pozwala dopisywać rekordów na koniec istniejącego pliku

TAK

NIE

020.txt

X01

Sekwencyjny dostęp do pliku nie powinien być używany - jest implementowany jedynie z przyczyn historycznych

TAK

NIE

021.txt

X01

Sekwencyjny dostęp do pliku pozwala czytać rekordy w dowolnej kolejności dzięki operacji lseek

TAK

NIE

022.txt

X10

Sekwencyjny dostęp do pliku wyklucza powstawanie "dziur" w plikach, w których nie ma przydzielonych bloków

TAK

NIE

023.txt

X10

Sekwencyjny dostęp do pliku ułatwia grupowanie operacji dyskowych przez system operacyjny

TAK

NIE

024.txt

X01

Sekwencyjny dostęp do pliku powinien być używany jedynie przy operacjach na plikach specjalnych, odpowiadających urządzeniom znakowym

TAK

NIE

025.txt

X01

Sekwencyjny dostęp do pliku szereguje dostęp wielu procesów do pliku

TAK

NIE

026.txt

X01

Bezpośredni==swobodny dostęp do pliku wymaga zmiany kolejności rekordów w pliku w celu optymalizacji czasu dostępu

TAK

NIE

027.txt

X10

Bezpośredni==swobodny dostęp do pliku pozwala na występowanie plików nieciągłych, wewnątrz których mogą wystąpić niezaalokowane obszary

TAK

NIE

028.txt

X10

Bezpośredni==swobodny dostęp do pliku jest możliwy tylko w plikach z rekordami stałej długości

TAK

NIE

029.txt

X10

Bezposredni==swobodny dostep do pliku pozwala na wystepowanie plikow nieciaglych, z niezaalokowanymi dziurami

TAK

NIE

030.txt

X10

Bezposredni==swobodny dostep do pliku pozwala czytac rekordy w dowolnej kolejnosci dzięki operacji lseek

TAK

NIE

031.txt

X01

Bezposredni==swobodny dostep do pliku pozwala na zapis fragmentow roznych plikow do wspolnego bloku alokacji

TAK

NIE

032.txt

X01

Bezposredni==swobodny dostep do pliku szereguje dostep wielu procesow do pliku

TAK

NIE

033.txt

X01

Bezposredni==swobodny dostep do pliku blokuje rownoczesny dostep wielu procesow do pliku

TAK

NIE

034.txt

X01

W unixie wystepuje katalog, dostepny przez sciezke ../ jest to katalog roboczy procesu

TAK

NIE

035.txt

X10

W unixie wystepuje katalog, dostepny przez sciezke ../ jest to katalog nadrzedny

TAK

NIE

036.txt

X01

W unixie występuje katalog, dostępny przez ścieżkę ../ jest to katalog główny, jeden dla całego systemu

TAK

NIE

037.txt

X10

W unixie występuje katalog, dostępny przez ścieżkę ../ każdy proces może mieć swój taki katalog

TAK

NIE

038.txt

X01

W unixie występuje katalog, dostępny przez ścieżkę ../ proces musi mieć prawa zapisu w tym katalogu, aby mógł się wykonywać

TAK

NIE

039.txt

X01

Przydzielanie ciągłych miejsc na dysku jest algorytmem optymalnym, nie dającym się zaimplementować w praktyce

TAK

NIE

040.txt

X10

Przydzielanie ciągłych miejsc na dysku charakteryzuje się fragmentacją zewnętrzną przestrzeni dysku

TAK

NIE

041.txt

X01

Przydzielanie ciągłych miejsc na dysku wymaga zapisu tablicy indeksowej na dysku

TAK

NIE

042.txt

X01

Przydzielanie indeksowych miejsc na dysku jest algorytmem optymalnym, nie dającym się zaimplementować w praktyce

TAK

NIE

043.txt

X01

Przydział indeksowy miejsca na dysku charakteryzuje się fragmentacją zewnętrzną przestrzeni dysku

TAK

NIE

044.txt

X10

Przydział indeksowy miejsca na dysku wymaga zapisu tablicy indeksowej na dysku

TAK

NIE

045.txt

X01

Przydział listowy miejsca na dysku jest algorytmem optymalnym, nie dającym się zaimplementować w praktyce

TAK

NIE

046.txt

X01

Przydział listowy miejsca na dysku charakteryzuje się brakiem fragmentacji wewnętrznej przestrzeni dysku

TAK

NIE

047.txt

X01

Przydział listowy miejsca na dysku wymaga zapisu tablicy listowej na dysku

TAK

NIE

048.txt

X01

Planowanie krótkoterminowe wymaga architektury wieloprotokółowej

TAK

NIE

049.txt

X01

Planowanie krótkoterminowe bezpośrednio decyduje o wyborze procesów, które należy wczytać do pamięci

TAK

NIE

050.txt

X01

Planowanie krótkoterminowe określa procesy, które należy zablokować

TAK

NIE

051.txt

X01

Planowanie średnioterminowe wymaga architektury wieloprocesorowej

TAK

NIE

052.txt

X10

Planowanie średnioterminowe bezpośrednio decyduje o wyborze procesów, które należy wczytać do pamięci

TAK

NIE

053.txt

X10

Planowanie średnioterminowe określa procesy, które należy zablokować

TAK

NIE

054.txt

X10

Planowanie średnioterminowe polega na wyborze procesu, który ma zostać usunięty z pamięci

TAK

NIE

055.txt

X10

Planowanie średnioterminowe to wybór zadania, które w pierwszej kolejności powinno zostać wczytane do pamięci

TAK

NIE

056.txt

X01

Planowanie średnioterminowe musi być zaimplementowane w każdym wieloprogramowym SO

TAK

NIE

057.txt

X01

Planowanie długoterminowe wymaga architektury wieloprocessorowej

TAK

NIE

058.txt

X10

Planowanie długoterminowe bezpośrednio decyduje o wyborze procesów, które należy wczytać do pamięci

TAK

NIE

059.txt

X10

Planowanie długoterminowe określa procesy, które należy utworzyć

TAK

NIE

060.txt

X01

“Spooling” to technika przydzielania dodatkowych buforów procesom uprzywilejowanym

TAK

NIE

061.txt

X10

“Spooling” to technika kopiowania w tło danych z nośnika papierowego na szybszy magnetyczny

TAK

NIE

062.txt

X01

“Spooling” to technika programowania, zmniejszająca pamięć wymagana do wykonania programu

TAK

NIE

063.txt

X01

“Spooling” eliminuje fragmentację zewnętrznych dysków przez przesuwanie plików w trakcie normalnej pracy komputera (w tle)

TAK

NIE

064.txt

X01

"Spooling" eliminuje fragmentacje zewnętrznych dysków przez przesuwanie plików, przy czym uruchamiany jest przy

TAK

NIE

065.txt

X01

"Spooling" wyłączonej wszystkich innych procesach

TAK

NIE

066.txt

X01

"Spooling" to metoda buforowania dysku bazująca na tworzeniu dużych puli buforów systemowych

TAK

NIE

067.txt

X01

"Spooling" to metoda buforowania dysku w szybkich pamięciach wbudowanych do sterownika

TAK

NIE

068.txt

X10

"Spooling" polega na zapisie danych wyjściowych z programu na szybkie nośniki magnetyczne i drukowaniu ich w tle po zakończeniu programu

TAK

NIE

069.txt

X10

W Unixie występuje katalog, dostępny przez ścieżkę ./ jest to katalog roboczy procesu

TAK

NIE

070.txt

X01

W Unixie występuje katalog, dostępny przez ścieżkę ./ jest to katalog nadrzędny

TAK

NIE

071.txt

X01

W unixie występuje katalog, dostępny przez ścieżkę ./ jest to katalog główny, jeden dla całego systemu

TAK

NIE

072.txt

X10

W unixie występuje katalog, dostępny przez ścieżkę ./ każdy proces może mieć swój taki katalog

TAK

NIE

073.txt

X01

W unixie występuje katalog, dostępny przez ścieżkę ./ proces musi mieć prawa zapisu w tym katalogu, aby mógł się wykonywać

TAK

NIE

074.txt

X10

Katalog główny systemu plików w unixie jest plikiem zawierającym rekordy z nazwami i numerami inode'ów należących do niego plików

TAK

NIE

075.txt

X01

Katalog główny systemu plików w unixie jest plikiem zawierającym rekordy z nazwami, innymi atrybutami i tablicami indeksowymi należących do niego plików

TAK

NIE

076.txt

X01

Katalog główny systemu plików w unixie nie ma nazwy, definiowany jest przez zapisanie numeru jego inode'u w superbloku

TAK

NIE

077.txt

X01

Katalog główny systemu plików w unixie nie ma nazwy, zajmuje zawsze to samo miejsce na partycji dysku (w systemie plików)

TAK

NIE

078.txt

X01

Katalog glówny systemu plików w unixie ma ograniczona wielkosc, poniewaz umieszczony jest w calosci w superbloku

TAK

NIE

079.txt

X01

Katalog glówny systemu plików w unixie nie zawiera pozycji .. (normalnie wskazujacej na katalog nadrzedny)

TAK

NIE

080.txt

X01

Katalog glówny systemu plików w unixie nie zawiera pozycji . i .. (normalnie wskazujacych na katalog biezacy i nadrzedny)

TAK

NIE

081.txt

X01

Katalog glówny systemu plików w unixie zawiera stala liczbe pozycji, ktorej nie mozna zmienic bez rekompilacji jadra systemu

TAK

NIE

082.txt

X01

Katalog glówny systemu plików w unixie musi miec taka sama zawartosc, jak katalog do ktorego jest montowany

TAK

NIE

083.txt

X10

Katalog biezacy w unixie jest plikiem zawierajacym rekordy z nazwami i numerami inode'ow nalezacych do niego plikow

TAK

NIE

084.txt

X01

Katalog biezacy w unixie ma ograniczona wielkosc, poniewaz umieszczony jest w calosci w superbloku

TAK

NIE

085.txt

X01

Katalog biezacy w unixie nie zawiera pozycji . i ..

TAK

NIE

086.txt

X01

Katalog biezacy w unixie uzytkwnik nie moze zmienic go, jezeli nie ma uprawnień root

TAK

NIE

087.txt

X10

Powloka systemu operacyjnego okresla dostepny zestaw i skladnie komend sterujacych

TAK

NIE

088.txt

X01

Powloka systemu operacyjnego okresla dostepny zestaw i skladnie komend zewnetrznych

TAK

NIE

089.txt

X10

Powloka systemu operacyjnego okresla dostepny zestaw i skladnie komend wewnetrznych

TAK

NIE

090.txt

X10

Powloka systemu operacyjnego okresla zmienne srodowiskowe dla uruchamianych komend

TAK

NIE

091.txt

X10

Powloka systemu operacyjnego okresla jezyk programowania skryptow systemowych

TAK

NIE

092.txt

X10

Powłoka systemu operacyjnego pozwala na edytowanie linii komendy

TAK

NIE

093.txt

X10

Powłoka systemu operacyjnego pozwala na przeglądanie historii wykonywanych komend

TAK

NIE

094.txt

X01

Listy kontroli dostępu ACL w windows NT służą do określania warunków logowania użytkowników do systemu

TAK

NIE

095.txt

X10

Listy kontroli dostępu ACL w windows NT służą do określania warunków dostępu do plików dla różnych użytkowników

TAK

NIE

096.txt

X10

Listy kontroli dostępu ACL w windows NT pozwalają definiować grupy użytkowników

TAK

NIE

097.txt

X01

Listy kontroli dostępu ACL w windows NT pozwalają definiować klucze dostępu do zasobów

TAK

NIE

098.txt

X10

Listy kontroli dostępu ACL w windows NT umożliwiają zdefiniowanie obiektów, do których dostęp musi być odnotowany

TAK

NIE

099.txt

X01

Plik MFT (Master file table) w systemie plików NTFS służy do określania uprawnień dostępu do plików

TAK

NIE

100.txt

X01

Plik MFT (Master file table) w systemie plików NTFS służy do określania warunków logowania użytkowników do systemu

TAK

NIE

101.txt

X10

Plik MFT (Master file table) w systemie plików NTFS zawiera listy ACL wszystkich plików

TAK

NIE

102.txt

X01

Plik MFT (Master file table) w systemie plików NTFS zawiera kopie boot sektora

TAK

NIE

103.txt

X10

Plik MFT (Master file table) w systemie plików NTFS zawiera opis wszystkich plików w systemie

TAK

NIE

104.txt

X01

Plik MFT (Master file table) w systemie plików NTFS zawiera tablice partycji dysku

TAK

NIE

105.txt

X01

Tablica FAT w systemie MS-DOS zawiera osobną pozycję dla każdego rekordu fizycznego (sektora) w partycji dysku

TAK

NIE

106.txt

X10

Tablica FAT w systemie MS-DOS zawiera informacje o defektach nośnika w pamięci dyskowej

TAK

NIE

107.txt

X01

Tablica FAT w systemie MS-DOS pozwala przydzielać rekordy rezerwowe w miejsce uszkodzonych

TAK

NIE

108.txt

X10

Tablica FAT w systemie MS-DOS zastępuje mapę zajętości dysku

TAK

NIE

109.txt

X01

Semafory binarne służą do wyznaczania czasu w systemach czasu rzeczywistego

TAK

NIE

110.txt

X10

Semafory binarne służą do synchronizacji procesów współbieżnych

TAK

NIE

111.txt

X01

Semafory binarne służą do blokowania dostępu do dysku w trybie binarnym (nie znakowym)

TAK

NIE

112.txt

X01

Semafory binarne są to struktury danych do modelowania ruchu ulicznego

TAK

NIE

113.txt

X10

Semafor binarny to dane jednobitowe, które można testować, ustawiać i zerować

TAK

NIE

114.txt

X01

Semafor binarny to dane całkowitoliczbowe z zakresu $0 - 2n$ ($n > 1$) z niepodzielnymi operacjami p i v

TAK

NIE

115.txt

X10

Semafor binarny można używać do rozwiązywania dowolnych problemów synchronizacji procesów

TAK

NIE

116.txt

X10

W semaforach dykstry operacja czekaj na tych semaforach zawsze usypia procesy

TAK

NIE

117.txt

X01

W semaforach dykstry operacja sygnalizuj na tych semaforach może usypiać procesy

TAK

NIE

118.txt

X01

Semafor dykstry to mechanizm ograniczającym dostęp do plików

TAK

NIE

119.txt

X10

Semafor dykstry służy do synchronizacji procesów

TAK

NIE

120.txt

X10

Semafor dykstry służy do synchronizacji procesów współbieżnych

TAK
NIE

121.txt
X01
Semafor Dijkstra to struktura danych do modelowania ruchu ulicznego
TAK
NIE

122.txt
X10
Semafor Dijkstra to mechanizm synchronizujący dostęp do plików
TAK
NIE

123.txt
X10
Semafor Dijkstra to dane całkowitoliczbowe, na których zdefiniowane są operacje P i V
TAK
NIE

124.txt
X01
Semafor Dijkstra to flagi dwustanowe, na których zdefiniowane są nieprzerwane operacje P i V
TAK
NIE

125.txt
X10
Semafor Dijkstra można używać do rozwiązywania dowolnych problemów synchronizacji procesów
TAK
NIE

126.txt
X01
Semafor Dijkstra w ich implementacji stosuje się pętle gorącego czekania
TAK
NIE

127.txt
X01
Kontekst procesu określa wszystkie zasoby, z których może proces korzystać
TAK
NIE

128.txt

X01

Kontekst procesu określa wszystkie procesy, z którymi może się komunikować

TAK

NIE

129.txt

X10

Kontekst procesu określa zawartość rejestrów, z których proces może korzystać

TAK

NIE

130.txt

X10

Kontekst procesu obejmuje informacje konieczne do wznowienia procesu zablokowanego

TAK

NIE

131.txt

X01

Kontekst procesu obejmuje informacje konieczne do ustalenia procesu nadrzędnego

TAK

NIE

132.txt

X01

Kontekst procesu jest to obraz pamięci procesu w pewnej chwili

TAK

NIE

133.txt

X01

Kontekst procesu jest to między innymi obraz pamięci procesu w chwili wstrzymania

TAK

NIE

134.txt

X10

Kontekst procesu obejmuje informacje konieczne do wznowienia procesu wstrzymanego

TAK

NIE

135.txt

X01

Kontekst procesu jest to obraz plików otwartych przez proces w pewnej chwili

TAK

NIE

136.txt

X01

Kontekst procesu jest to obraz pamięci, rejestrów i zmiennych systemowych procesu w pewnej chwili

TAK

NIE

137.txt

X01

Kontekst procesu jest składowany gdy proces się kończy (wykonuje exit)

TAK

NIE

138.txt

X01

Kontekst procesu jest odtwarzany przy każdym przełączeniu procesów

TAK

NIE

139.txt

X01

Kontekst procesu opisuje zależność pomiędzy współbieżnymi procesami

TAK

NIE

140.txt

X10

Procesy charakteryzują się sekwencyjnym wykonywaniem operacji

TAK

NIE

141.txt

X01

Procesy charakteryzują się kontekstem, który określa jakie procesy są wykonywane współbieżnie

TAK

NIE

142.txt

X10

Procesy charakteryzują się złożonością obliczeniową, wyrażoną np. przez liczbę pętli, skoków i rozgałęzień (Nie wiadomo do końca)

TAK

NIE

143.txt

X10

Procesy charakteryzują się możliwością przerywania i wznowiania wykonania (według składowanego kontekstu)

TAK

NIE

144.txt

X01

Proces może przejść do stanu gotowości po wyczerpaniu kwantu czasu aktywności (w systemie z algorytmem karuzelowym)

TAK

NIE

145.txt

X01

Proces może przejść do stanu gotowości w wyniku wykonania funkcji exit (w unixie)

TAK

NIE

146.txt

X10

Proces może przejść do stanu gotowości bezpośrednio ze stanu zablokowania

TAK

NIE

147.txt

X10

Proces może przejść do stanu zablokowania (uspiania) po wyczerpaniu kwantu czasu aktywności (w systemie z algorytmem karuzelowym)

TAK

NIE

148.txt

X01

Proces może przejść do stanu zablokowania (uspiania) w wyniku wykonania funkcji exit (w unixie)

TAK

NIE

149.txt

X01

Proces może przejść do stanu zablokowania (uspiania) bezpośrednio ze stanu gotowości

TAK

NIE

150.txt

X10

Ochrona pamieci w systemach operacyjnych ogranicza dostep procesow do pamieci systemu operacyjnego

TAK

NIE

151.txt

X10

Ochrona pamieci w systemach operacyjnych ogranicza dostep procesow do pamieci im nie przydzielonej

TAK

NIE

152.txt

X10

Ochrona pamieci w systemach operacyjnych zapewnia, ze inne procesy nie maja dostepu do pamieci procesu

TAK

NIE

153.txt

X10

Ochrona pamieci w systemach operacyjnych zapewnia, ze procesy uzytkowe nie moga zmodyfikowac obrazu pamieci systemu oper.

TAK

NIE

154.txt

X10

Ochrona pamieci w systemach operacyjnych bazuje na przekszaltaniu adresow logicznych na fizyczne, kontrolowanym przez sO

TAK

NIE

155.txt

X01

Ochrona pamieci w systemach operacyjnych bazuje na szyfrowaniu zawartosci pamieci

TAK

NIE

156.txt

X01

Ochrona pamieci w systemach operacyjnych okresla prawa dostepu procesow do plikow roznym
wlasoscieli

TAK

NIE

157.txt

X01

Ochrona pamieci w systemach operacyjnych okresla prawa dostepu procesow do folderow poczty
elektronicznej

TAK

NIE

158.txt

X10

Stronicowanie na zadanie to podstawowa technika zwiekszania pamieci widzianej przez programy

TAK

NIE

159.txt

X01

Stronicowanie na zadanie to podstawowa technika przyspieszenia dostepu do pamieci

TAK

NIE

160.txt

X01

Stronicowanie na zadanie polega na skladowaniu na dysk calego obrazu pamieci zablokowanego
procesu

TAK

NIE

161.txt

X01

Stronicowanie na zadanie polega na buforowaniu stron pamieci w pamieci podrecznej cache

TAK

NIE

162.txt

X10

Stronicowanie na zadanie wymaga dodatkowych pol w tablicy stron (np. flagi waznosci)

TAK

NIE

163.txt

X10

Stronicowanie na zadanie wymaga utrzymywania w pamięci tablicy stron

TAK

NIE

164.txt

X10

Stronicowanie pamięci polega na przydziale do programu ramek pamięci o stałej wielkości

TAK

NIE

165.txt

X01

Stronicowanie pamięci polega na przydziale do programu segmentów pamięci dowolnej wielkości

TAK

NIE

166.txt

X01

Stronicowanie pamięci wymaga tworzenia obrazu pamięci w obszarze wymiany na dysku

TAK

NIE

167.txt

X01

Stronicowanie pamięci pozwala implementować liniowy model adresów fizycznych w programie

TAK

NIE

168.txt

X10

Stronicowanie pamięci pozwala implementować liniowy model adresów logicznych w programie

TAK

NIE

169.txt

X01

Pamięć stronicowania charakteryzuje się tym, że adresy logiczne są tworzone jako suma adresu bazowego i przesunięcia

TAK

NIE

170.txt

X10

Pamięć stronicowania charakteryzuje się tym, że adresy fizyczne są sklejaniem pary liczb określających numer ramki stałej długości oraz przesunięcia (offset) w obrębie ramki

TAK
NIE

171.txt

X10

Pamięć stronicowania charakteryzuje się tym, że adres logiczny jest złożeniem numeru strony i przesunięcia względem jej początku

TAK
NIE

172.txt

X10

Pamięć stronicowania charakteryzuje się tym, że jeżeli strona nie ma przydzielonej ramki fizycznej, to występuje błąd strony; brak przydziału jest

TAK
NIE

173.txt

X01

Pamięć stronicowania charakteryzuje się tym, że oznaczony w tablicy stron flaga poprawności (valid)

TAK
NIE

174.txt

X01

Pamięć stronicowania charakteryzuje się tym, że jeżeli strona nie ma przydzielonej ramki fizycznej, to występuje błąd strony; brak przydziału jest oznaczony w tablicy stron bitem ochrony (access control)

TAK
NIE

175.txt

X01

Pamięć stronicowania charakteryzuje się tym, że jeżeli strona nie ma przydzielonej ramki fizycznej, to występuje błąd strony; brak przydziału jest oznaczony w tablicy stron bitem modyfikacji (modified flag)

TAK
NIE

176.txt

X01

Pamięć stronicowania charakteryzuje się tym, że jest technika pozwalająca przydzielać programom więcej pamięci, niż jest fizycznie zainstalowane w systemie

TAK
NIE

177.txt

X10

Pamięć stronicowania charakteryzuje się tym, że jest technika pozwalająca przydzielać programom pamięć blokami stałej wielkości (stronami)

TAK

NIE

178.txt

X10

Pamięć stronicowania charakteryzuje się tym, że adres logiczny jest przekształcany na adres fizyczny przez odczytanie numeru ramki z tablicy stron, z pozycji odpowiadającej numerowi strony

TAK

NIE

179.txt

X01

Pamięć stronicowania charakteryzuje się tym, że jest technika pozwalająca przydzielać programom pamięć blokami zmiennej wielkości

TAK

NIE

180.txt

X10

Pamięć stronicowania charakteryzuje się tym, że eliminuje fragmentację zewnętrzną pamięci

TAK

NIE

181.txt

X01

Pamięć stronicowania charakteryzuje się tym, że eliminuje fragmentację wewnętrzną pamięci

TAK

NIE

182.txt

X01

Pamięć stronicowania charakteryzuje się tym, że musi być realizowana z wykorzystaniem pamięci asocjacyjnej na tablicy stron

TAK

NIE

183.txt

X10

Pamięć stronicowania charakteryzuje się tym, że powoduje, że programy mają nieliniową przestrzeń adresową (fizyczna tak, logiczna nie)

TAK

NIE

184.txt

X01

Pamięć stronicowania charakteryzuje się tym, że wymaga implementacji tablicy segmentów (sprzętowo lub programowo)

TAK

NIE

185.txt

X01

Blokada systemu występuje, gdy dwa procesy próbują równocześnie pisać do tego samego pliku

TAK

NIE

186.txt

X01

Blokada systemu występuje, gdy dwa procesy próbują równocześnie pisać do tego gniazda sieciowego

TAK

NIE

187.txt

X01

Blokada systemu występuje tylko w systemach operacyjnych napisanych z błędami

TAK

NIE

188.txt

X10

Blokada systemu występuje tylko w systemach operacyjnych wieloprogramowych

TAK

NIE

189.txt

X10

Blokadę systemu można usunąć przez wymuszenie zakończenia wszystkich procesów uspionych

TAK

NIE

190.txt

X01

Blokadę systemu można usunąć przez czasowe zablokowanie dostępu użytkowników do systemu

TAK

NIE

191.txt

X10

Warunki konieczne do wystąpienia blokady to występowanie zasobów, które nie mogą być równocześnie wykorzystywane przez wiele procesów

TAK

NIE

192.txt

X01

Warunki konieczne do wystąpienia blokady to brak priorytetów w dostępie do zasobów

TAK

NIE

193.txt

X01

Warunki konieczne do wystąpienia blokady to przełączanie procesów według algorytmu planowania z wywłaszczaniem

TAK

NIE

194.txt

X10

Warunki konieczne do wystąpienia blokady to przełączanie procesów według algorytmu bez wywłaszczania

TAK

NIE

195.txt

X10

Warunki konieczne do wystąpienia blokady to przetrzymywanie i oczekiwanie, tzn. pozwolenie procesom nie będącym aktywnymi ani gotowymi na zachowanie przydziału zasobu

TAK

NIE

196.txt

X10

Warunki konieczne do wystąpienia blokady to brak mechanizmu, który umożliwiałby procesom przyjęcie zasobów przydzielonych innym

TAK

NIE

197.txt

X10

Warunki konieczne do wystapienia blokady to wystepowanie zasobow, ktore nie moga byc
wspoldzielone
TAK
NIE

198.txt
X10
Warunki konieczne do wystapienia blokady to brak mechanizmu umozliwiajacego wywlaszczanie
zasobow od procesow
TAK
NIE

199.txt
X01
Warunki konieczne do wystapienia blokady to okreslaja, jak pisac programy wielowatkowe aby
unikac bloakd
TAK
NIE

200.txt
X01
Warunki konieczne do wystapienia blokady to okreslaja, jak pisac programy wieolprocesowe aby
unikac bloakd
TAK
NIE

201.txt
X01
Warunki konieczne do wystapienia blokady to okreslaja, kiedy proces moze przejsc do stanu
zablokowania
TAK
NIE

202.txt
X10
Warunki konieczne do wystapienia blokady to mozliwosc zawieszania procesow z przydzielonymi
pewnymi zasobami
TAK
NIE

203.txt
X10
Funkcja systemowa wait w unixie zawiesza wykonywanie procesu w oczekiwaniu na zakonczenie
procesu potomnego
TAK
NIE

204.txt

X01

Funkcja systemowa wait w unixie zawiesza wykonywanie procesu przez zadany argumentem czas

TAK

NIE

205.txt

X01

Funkcja systemowa wait w unixie może być wykonywana tylko przez procesy daemony

TAK

NIE

206.txt

X01

Funkcja systemowa wait w unixie może być wykonywana tylko przez proces z uprawnieniami root

TAK

NIE

207.txt

X10

Funkcja systemowa wait w unixie umożliwia czekanie na zakończenie operacji sieciowych

TAK

NIE

208.txt

X10

Funkcja systemowa wait w unixie umożliwia czekanie na zakończenie operacji we/wy

TAK

NIE

209.txt

X10

Technika opóźnionego zapisu na dysk w systemie unix jest metodą sieciową

TAK

NIE

210.txt

X01

Technika opóźnionego zapisu na dysk w systemie unix jest metodą synchronizacji zapisów na dysku

TAK

NIE

211.txt

X10

Technika opóźnionego zapisu na dysk w systemie unix polega na wprowadzeniu petli opóźniającej między kolejnymi rozkazami zapisu

TAK

NIE

212.txt

X01

Technika opóźnionego zapisu na dysk w systemie unix polega na zapisie bufora na dysk, w chwili jego zwalniania

TAK

NIE

213.txt

X01

Technika opóźnionego zapisu na dysk w systemie unix może być wybiórczo stosowana przez programowe włączanie/wyłączanie funkcja fctl

TAK

NIE

214.txt

X01

Technika opóźnionego zapisu na dysk w systemie unix może być wyłączana komenda fctl

TAK

NIE

215.txt

X01

Buforowanie operacji dyskowych w systemie unix jest ograniczone do urządzeń z niewymiennym nośnikiem (np. twardych dysków); dostęp do plików na dyskietkach nie może być buforowany

TAK

NIE

216.txt

X10

Buforowanie operacji dyskowych w systemie unix redukuje czas dostępu do plików oraz liczbę fizycznych operacji we/wy

TAK

NIE

217.txt

X10

Buforowanie operacji dyskowych w systemie unix utrudnia równoczesny dostęp do rekordów pliku przez wiele procesów (recordlocking)

TAK

NIE

218.txt

X01

Buforowanie operacji dyskowych w systemie unix wymaga od programisty uzywania operacji fflush aby miec pewnosc, ze zmiany w pliku zostana natychmiast zapisane na dysk

TAK

NIE

219.txt

X10

Sygnały w systemie unix wykorzystywane sa do synchronizacji procesow

TAK

NIE

220.txt

X10

Sygnały w systemie unix wykorzystywane sa m.in. do zabijania procesow

TAK

NIE

221.txt

X10

Sygnały w systemie unix moga byc adresowane do wszystkich procesow uzytkownika o znanym uid

TAK

NIE

222.txt

X01

Sygnały w systemie unix moga byc adresowane do wszystkich procesow uspionych w systemie

TAK

NIE

223.txt

X10

Sygnały w systemie unix mozna definiowac wlasne funkcje obslugi niektorych z nich

TAK

NIE

224.txt

X01

Sygnały w systemie unix proces mozna zaprogramowac tak, aby ignorowal wszystkie sygnaly przychodzace

TAK

NIE

225.txt

X01

Sygnały w systemie unix wysyłane mogą być przez każdy proces przy pomocy funkcji signal

TAK

NIE

226.txt

X10

Sygnały w systemie unix wysyłane mogą być przez każdy proces przy pomocy funkcji kill

TAK

NIE

227.txt

X01

Fragmentacja wewnętrzna dysku jest wynikiem niedopasowania rozmiaru plików do wielkości obszarów wolnych (powstających przy kasowaniu plików)

TAK

NIE

228.txt

X10

Fragmentacja wewnętrzna dysku jest wynikiem niedopasowania rozmiaru plików do wielokrotności wielkości bloków alokacji

TAK

NIE

229.txt

X01

Fragmentacja wewnętrzna dysku jest wynikiem przydziału miejsca na dysku porcjami zmiennej wielkości

TAK

NIE

230.txt

X01

Fragmentacja wewnętrzna dysku może być wyeliminowana przez program defragmentujący

TAK

NIE

231.txt

X10

Fragmentacja wewnętrzna dysku jest wynikiem pozostawiania niewykorzystanych fragmentów w ostatnich blokach plików

TAK

NIE

232.txt

X10

Fragmentacja wewnętrznego dysku jest wynikiem przydziału miejsca na dysku porcjami stałej wielkości

TAK

NIE

233.txt

X01

Fragmentacja wewnętrznego dysku jest wynikiem braku konserwacji systemu plików - może być wyeliminowana przez program defragmentujący

TAK

NIE

234.txt

X01

Fragmentacja wewnętrznego dysku jest wynikiem braku konserwacji systemu plików (brak defragmentacji)

TAK

NIE

235.txt

X01

Fragmentacja wewnętrznego dysku jest wynikiem błędów programowych np. nie zamknięcia pliku przed zakończeniem programu

TAK

NIE

236.txt

X01

Fragmentacja wewnętrznego dysku powoduje lepsze wykorzystanie miejsca w pamięci przez procesy

TAK

NIE

237.txt

X10

Fragmentacja wewnętrznego dysku jest wynikiem przydziału pamięci porcjami stałej wielkości (stronami)

TAK

NIE

238.txt

X01

Fragmentacja wewnętrznego dysku występuje jedynie w systemach z pamięcią wirtualną

TAK

NIE

239.txt

X01

Fragmentacja wewnętrzna dysku jest wynikiem niedopasowania rozmiaru segmentów pamięci do wielokrotności wielkości ramki

TAK

NIE

240.txt

X01

Fragmentacja wewnętrzna dysku może być wyeliminowana przez relokację programów

TAK

NIE

241.txt

X01

Fragmentacja wewnętrzna dysku jest wynikiem błędów programowych np. nie zwalniania pamięci przed zakończeniem programu

TAK

NIE

242.txt

X01

Fragmentacja zewnętrzna pamięci jest wynikiem niedopasowania rozmiaru segmentów pamięci do wielokrotności wielkości ramki

TAK

NIE

243.txt

X10

Fragmentacja zewnętrzna pamięci może być wyeliminowana przez relokację programów

TAK

NIE

244.txt

X01

Fragmentacja zewnętrzna pamięci jest wynikiem błędów programowych np. nie zwalniania pamięci przed zakończeniem programu

TAK

NIE

245.txt

X10

Algorytm przełączania procesów w systemie unix jest algorytmem ze stałym priorytetem zewnętrznym, zadawanym komendą/funkcją nice

TAK

NIE

246.txt

X01

Algorytm przełączania procesów w systemie unix jest algorytmem bez wyłączenia

TAK

NIE

247.txt

X01

Algorytm przełączania procesów w systemie unix jest algorytmem z wyłączeniem procesów systemowych i użytkowych

TAK

NIE

248.txt

X10

Algorytm przełączania procesów w systemie unix jest algorytmem, w którym wyłączone są tylko procesy użytkownika

TAK

NIE

249.txt

X10

Algorytm przełączania procesów w systemie unix jest algorytmem priorytetowym, w którym priorytet jest wewnętrznie zmieniany w trakcie wykonywania procesu

TAK

NIE

250.txt

X10

Algorytm przełączania procesów w systemie unix jest algorytmem z wyłączeniem

TAK

NIE

251.txt

X01

Algorytm przełączania procesów w systemie unix opiera się na priorytecie statycznym, wyliczanym w oparciu o wykorzystanie procesora przez proces

TAK

NIE

252.txt

X10

Algorytm przełączania procesów w systemie unix opiera się na priorytecie dynamicznym, obniżanym dla procesów intensywnych obliczeniowo, a podwyższanym dla procesów interakcyjnych

TAK

NIE

253.txt

X10

Algorytm przełączania procesów w systemie unix pozwala modyfikować priorytet procesu przy pomocy funkcji nice

TAK

NIE

254.txt

X01

Algorytm karuzelowy przełączania procesów jest algorytmem bez wyłączenia

TAK

NIE

255.txt

X10

Algorytm karuzelowy przełączania procesów jest algorytmem z wyłączeniem

TAK

NIE

256.txt

X01

Algorytm karuzelowy przełączania procesów przełącza procesy według dynamicznie zmienianego priorytetu

TAK

NIE

257.txt

X01

Algorytm karuzelowy przełączania procesów jest najczęściej wykorzystywanym algorytmem w różnych wersjach systemu operacyjnego unix

TAK

NIE

258.txt

X10

Algorytm karuzelowy przełączania procesów polega na uruchamianiu procesów na stały kwant czasu i wymuszaniu przełączenia kontekstu, jeżeli proces nie zwolni procesora w tym czasie

TAK

NIE

259.txt

X01

Algorytm karuzelowy przełączania procesów jest algorytmem optymalnym, w praktyce niemożliwym do zrealizowania

TAK

NIE

260.txt

X10

Algorytmy przełączania procesów z wywłaszczeniem mogą wykorzystywać priorytety stałe lub zmieniające się dynamicznie

TAK

NIE

261.txt

X01

Algorytmy przełączania procesów z wywłaszczeniem zwiększają współczynnik wykorzystania procesora w systemach wielodostępnych

TAK

NIE

262.txt

X10

Algorytmy przełączania procesów z wywłaszczeniem polegają na uruchamianiu procesów na stały kwant czasu i wymuszaniu przełączenia kontekstu, jeżeli proces nie zwolni procesora w tym czasie

TAK

NIE

263.txt

X01

Algorytm FIFO zastępowania stron pamięci gwarantuje najmniejszą możliwą liczbę błędów stron

TAK

NIE

264.txt

X01

Algorytm FIFO zastępowania stron pamięci gwarantuje najkrótszy czas dostępu do pamięci

TAK

NIE

265.txt

X01

Algorytm FIFO zastępowania stron pamięci zapewnia, że liczba błędów stron (dla tych samych procesów) nie zwiększy się, gdy doinstalujemy (udostępnimy) dodatkowe ramki pamięci fizycznej

TAK
NIE

266.txt

X10

Algorytm FIFO zastępowania stron pamięci polega na usuwaniu stron, które najdłużej przebywały w pamięci fizycznej

TAK

NIE

267.txt

X01

Algorytm FIFO zastępowania stron pamięci polega na usuwaniu stron, które najdawniej były używane

TAK

NIE

268.txt

X01

Algorytm IRU (najdawniej używana) zastępowania stron pamięci gwarantuje najmniejszą możliwą liczbę błędów stron

TAK

NIE

269.txt

X10

Algorytm IRU (najdawniej używana) zastępowania stron pamięci zapewnia, że liczba błędów stron (dla tych samych procesów) nie zwiększy się, gdy doinstalujemy (udostępnimy) dodatkowe ramki pamięci fizycznej

TAK

NIE

270.txt

X01

Algorytm IRU (najdawniej używana) zastępowania stron pamięci polega na usuwaniu stron, które najdłużej przebywały w pamięci fizycznej

TAK

NIE

271.txt

X01

Systemy wielodostępne wymagają architektury wieloprocessorowej

TAK

NIE

272.txt

X01

Systemy wielodostępne wymagają architektury jednoprosesorowej

TAK

NIE

273.txt

X10

Systemy wielodostępne pozwalają na wykonywanie wielu zadań współbieżnie

TAK

NIE

274.txt

X10

Systemy wielodostępne pozwalają na współbieżne przetwarzanie wielu procesów (jednego lub wielu użytkowników)

TAK

NIE

275.txt

X10

Systemy wielodostępne zapewniają dostęp wszystkim użytkownikom do identycznych danych

TAK

NIE

276.txt

X10

Systemy wielodostępne pozwalają na przesyłanie sobie poczty w obrębie systemu

TAK

NIE

277.txt

X01

Systemy wielodostępne wymuszają pracę wsadową na komputerze

TAK

NIE

278.txt

X01

Systemy wielodostępne wymagają wielu terminali podłączonych do jednego komputera

TAK

NIE

279.txt

X10

Systemy wielodostępne dzielą zadania na interakcyjne i wykonywane w tle

TAK
NIE

280.txt

X01

Dynamiczne ładowanie programu polega na wykorzystaniu bibliotek dołączanych w trakcie wykonywania programu

TAK
NIE

281.txt

X10

Dynamiczne ładowanie programu polega na ładowaniu nakładek programu, gdy są potrzebne

TAK
NIE

282.txt

X10

Dynamiczne ładowanie programu można zrealizować w systemie ze stronicowaniem na zadanie

TAK
NIE

283.txt

X10

Dynamiczne łączenie programu polega na wykorzystaniu bibliotek dołączanych w trakcie wykonywania programu

TAK
NIE

284.txt

X01

Dynamiczne łączenie programu polega na ładowaniu nakładek programu, gdy są potrzebne

TAK
NIE

285.txt

X10

Dynamiczne łączenie programu można zrealizować w systemie bez stronicowania na zadanie

TAK
NIE

286.txt

X10

Kryteriami rozwiązania problemu sekcji krytycznej jest wzajemne wykluczanie - procesy muszą mieć możliwość wzajemnego blokowania

TAK
NIE

287.txt

X10

Kryteriami rozwiązanie problemu sekcji krytycznej jest postęp - dostęp do sekcji nie może być zablokowany, jeżeli nie przebywa w niej inny

TAK
NIE

288.txt

X10

Kryteriami rozwiązanie problemu sekcji krytycznej jest ograniczone czekanie - każdy proces musi kiedyś dostać się do sekcji

TAK
NIE

289.txt

X01

Dolaczenia w systemie plikow (links) oznaczone sa symbolem / w sciezках dostępu

TAK
NIE

290.txt

X10

Dolaczenia w systemie plikow (links) pozwalaja tworzyć systemy plikow o strukturze niehierarchicznej

TAK
NIE

291.txt

X01

Dolaczenia w systemie plikow (links) sa wynikiem bledow w programie i polegaja na polaczeniu sie list alokacji odpowiadajacych roznym plikom w tablicy FAT

TAK
NIE

292.txt

X01

Dolaczenia w systemie plikow (links) sa wynikiem bledow w programie i powstaja przy wielokrotnym otwieraniu tych samych plikow do zapisu

TAK
NIE

293.txt

X10

Dolaczenia w systemie plikow (links) wiaza nazwy plikow z INODEíami w systemie unix

TAK

NIE

294.txt

X01

Dolaczenia w systemie plikow (links) tworzone sa przy otwieraniu plikow, usuwane przy zamykaniu

TAK

NIE

295.txt

X10

Dolaczenia w systemie plikow (links) moga byc symboliczne, gdy wskazuja sciezke dostepu do pliku wlasciwego, lub twarde, jezeli powstaja przez zdefiniowanie drugiej nazwy dla pliku

TAK

NIE

296.txt

X10

Przetwarzanie potokowe w unixie realizowane jest przez przekierowanie standardowego wyjscia komendy pierwszej do standardowego wejscia komendy drugiej

TAK

NIE

297.txt

X01

Przetwarzanie potokowe w unixie w powloce sh mozna je wywolac wypisujac dwie komendy w nawiasach okraglych, rozdzielone przecinkami

TAK

NIE

298.txt

X01

Przetwarzanie potokowe w unixie dostepne jest tylko w systemach wieloprocessorowych

TAK

NIE

299.txt

X01

Proces uspiony (zablokowany) oczekuje na zwolnienie procesora przez proces aktywny

TAK

NIE

300.txt

X10

Proces uspiony (zablokowany) realizuje operacje semaforowe

TAK

NIE

301.txt

X01

Proces uspiony (zablokowany) zawsze usuwany jest z pamieci operacyjnej (kopiowany do obszaru wymiany)

TAK

NIE

302.txt

X10

Stronicowanie na zadanie polega na wymianie stron w pamieci fizycznej, wczytywaniu stron, do ktorych wystepuja aktualnie odwolania

TAK

NIE

303.txt

X10

Stronicowanie na zadanie umozliwia zapisywanie do obszaru wymiany na dysku fragmentow kontekstow procesow, gdy brakuje pamieci fizycznej

TAK

NIE

304.txt

X10

Mechanizm spotkaniowy to technika synchronizacji dostepu do pamieci wspoldzielonej przez procesy

TAK

NIE

305.txt

X10

Mechanizm spotkaniowy wykorzystywany jest do przesylania danych pomiedzy procesami

TAK

NIE

306.txt

X01

Mechanizm spotkaniowy wykorzystywany jest do negocjacji dostepu do plikow

TAK

NIE

307.txt

X01

Warunek postępu w synchronizacji między procesami to zapewnienie, że dwa procesy nie znajdują się w tym samym czasie w sekcji krytycznej

TAK

NIE

308.txt

X10

Warunek postępu w synchronizacji między procesami to wymaganie, aby synchronizacja nie powodowała nieograniczonego w czasie zawieszenia procesu

TAK

NIE

309.txt

X01

Warunek postępu w synchronizacji między procesami zapewnia, że proces nie jest wywłaszczany, gdy znajduje się w sekcji krytycznej

TAK

NIE

310.txt

X01

Planowanie zadań ma na celu unikanie blokad

TAK

NIE

311.txt

X01

Planowanie zadań ma na celu synchronizowanie dostępu do zasobów

TAK

NIE

312.txt

X10

Planowanie zadań ma na celu wybór procesu do aktywacji

TAK

NIE

313.txt

X01

Funkcja systemowa fork w Unixie (Linuxie) tworzy nowy proces, realizujący program podany w argumencie do tej funkcji

TAK

NIE

314.txt

X01

Funkcja systemowa fork w unixie (Linuxie) tworzy nowy watek, realizujacy funkcje podana w argumentach

TAK

NIE

315.txt

X10

Funkcja systemowa fork w unixie (Linuxie) tworzy nowy proces, realizujacy ten sam program co proces-rodzic

TAK

NIE

316.txt

X01

Sygnały w systemie unix generowane sa wytlaczenie przez sterowniki urzadzzen zewnltrnych

TAK

NIE

317.txt

X10

Sygnały w systemie unix wysylane sa do pojedynczych procesow, grup procesow, procesow okreslonego uzytkownika lub do wszystkich procesow

TAK

NIE

318.txt

X01

Sygnały w systemie unix moga byc ignorowane przez procesy (z wyjatkiem sygnalu 9)

TAK

NIE

319.txt

X10

Pozycje katalogowe zawieraja nazwy plikow

TAK

NIE

320.txt

X10

Pozycje katalogowe zawieraja informacje o polozeniu pliku na dysku

TAK

NIE

321.txt

X01

Pozycje katalogowe zawieraja sume kontrolna pliku

TAK

NIE

322.txt

X01

Sciezka do pliku moze byc wzgledna, bezwzgledna lub symboliczna

TAK

NIE

323.txt

X01

Sciezka do pliku jest sciezka wzgledna okreslajaca polozenie pliku wzgledem katalogu domowego uzytkownika

TAK

NIE

324.txt

X10

Sciezka do pliku jest sciezka bezwzgledna okreslaca polozenie pliku wzgledem katalogu glownego

TAK

NIE

325.txt

X01

Sekcja krytyczna to technika synchronizacji dostepu do pamieci wspoldzielonej przez procesy

TAK

NIE

326.txt

X10

Sekcja krytyczna wykorzystywana jest do ochrony danych przed rownoczesna ich modyfikacja przez procesy wspolbiezne

TAK

NIE

327.txt

X01

Sekcja krytyczna wykorzystywana jest do negocjacji dostepu do semaforow

TAK

NIE

328.txt

X10

Wspolbieznosc w systemach operacyjnych realizowana jest przez przelaczanie procesow, tzn. wstrzymywanie procesow co pewien czas i pre innych

TAK

NIE

329.txt

X01

Wspolbieznosc w systemach operacyjnych w powloce sh mozna ja wywolac wypisujac dwie komendy w nawiasach okraglych, rozdzielone przecinkami

TAK

NIE

330.txt

X01

Wspolbieznosc w systemach operacyjnych dostepna jest tylko w systemach wieloprocessorowych

TAK

NIE

331.txt

X10

Proces gotowy oczekuje na zwolnienie procesora przez proces aktywny

TAK

NIE

332.txt

X01

Proces gotowy realizuje operacje drugoplanowe

TAK

NIE

333.txt

X01

Proces gotowy czasami usuwany jest z pamieci operacyjnej (kopiowany do obszaru wymiany)

TAK

NIE

334.txt

X01

Funkcja systemowa exec w unixie (Linuxie) tworzy nowy proces, realizujacy program podany w argumentach do tej funkcji

TAK

NIE

335.txt

X01

Funkcja systemowa exec w unixie (Linuxie) tworzy nowy watek, realizujacy funkcje podana w argumentach

TAK

NIE

336.txt

X10

Funkcja systemowa exec w unixie (Linuxie) modyfikuje kontekst wykonywanego procesu, korzystajac z pliku wykonywalnego podanego w argumentach

TAK

NIE

337.txt

X10

Sygnały w systemie unix generowane sa przez wykonanie operacji kill

TAK

NIE

338.txt

X01

Sygnały w systemie unix moga powodowac zakonczenie procesu

TAK

NIE

339.txt

X10

Sygnały w systemie unix moga byc wyslane do wielu procesow rownoczesnie

TAK

NIE

340.txt

X10

Dowiazania symboliczne sa to pliki specjalne, ktore wskazuja na inne obiekty w systemie plikow

TAK

NIE

341.txt

X10

Dowiazania symboliczne zawieraja informacje o polozeniu pliku na dysku

TAK

NIE

342.txt

X01

Dowiazania symboliczne w unixie musza miec rozszerzenie .lnk

TAK
NIE

343.txt

X01

Wzorce plikow sa mechanizmem definiowania dopuszczalnej zawartosci plikow

TAK
NIE

344.txt

X10

Wzorce plikow pozwalaja definiowac sciezki wyszukiwania komend w systemach operacyjnych

TAK
NIE

345.txt

X10

Wzorce plikow wykorzystuja znaki ? oraz * do okreslania dowolnych znakow w nazwie pliku

TAK
NIE

346.txt

X01

Funkcja systemowa execve w unixie(Linuxie) tworzy nowy proces, realizujacy program podany w argumentach do tej funkcji

TAK
NIE

347.txt

X01

Funkcja systemowa execve w unixie(Linuxie) tworzy nowy watek, realizujacy funkcje podana w argumentach

TAK
NIE

348.txt

X01

Funkcja systemowa execve w unixie(Linuxie) tworzy nowy proces, realizujacy ten sam program co proces-rodzic

TAK
NIE

349.txt

X10

Pozycje katalogowe w systemie unix zawieraja nazwy plikow

TAK
NIE

350.txt
X10
Pozycje katalogowe w systemie unix zawieraja date modyfikacji pliku
TAK
NIE

351.txt
X01
Pozycje katalogowe w systemie unix zawieraja rozszerzenie 3 - znakowe
TAK
NIE

352.txt
X10
Stan procesu jest pamietany w tablicy procesow
TAK
NIE

353.txt
X01
Stan procesu jest wyznaczany przy pierwszym uruchomieniu procesu i potem sie nie zmienia
TAK
NIE

354.txt
X10
Stan procesu jezeli jest aktywny, to proces jest wykonywany przez procesor
TAK
NIE

355.txt
X01
Sygnaly w systemie unix generowane sa wytlaczenie przez sterowniki urzadzen zewnetrznych
TAK
NIE

356.txt
X10
Sygnaly w systemie unix wysylane sa do pojedynczych procesow lub do grup procesow
TAK
NIE

357.txt

X10

Większość sygnałów w systemie unix może być ignorowana przez procesy

TAK

NIE

358.txt

X01

Kanał komunikacyjny z ograniczonym buforowaniem to technika synchronizacji dostępu do karty sieciowej przez procesy

TAK

NIE

359.txt

X10

Kanał komunikacyjny z ograniczonym buforowaniem wykorzystywany jest do przesyłania danych pomiędzy procesami

TAK

NIE

360.txt

X10

Kanał komunikacyjny z ograniczonym buforowaniem może spowodować uspienie procesu piszącego do kanału

TAK

NIE

361.txt

X10

Warunek wykluczania w synchronizacji między procesami to zapewnienie, że dwa procesy nie znajdują się w tym samym czasie w sekcji krytycznej

TAK

NIE

362.txt

X01

Warunek wykluczania w synchronizacji między procesami to wymaganie, aby synchronizacja nie powodowała nieograniczonego w czasie zawieszenia procesu

TAK

NIE

363.txt

X01

Warunek wykluczania w synchronizacji między procesami zapewnia, że proces nie jest wywłaszczany, gdy znajduje się w sekcji krytycznej

TAK
NIE

364.txt

X01

Funkcja systemowa signal w unixie (Linuxie) tworzy nowy proces

TAK

NIE

365.txt

X01

Funkcja systemowa signal w unixie (Linuxie) wysyla komunikat do standardowego strumienia bledow

TAK

NIE

366.txt

X01

Funkcja systemowa signal w unixie (Linuxie) wysyla sygnal do procesu lub grupy procesow

TAK

NIE

367.txt

X10

Algorytm karuzelowy zapewnia wszystkim procesom ten sam priorytet

TAK

NIE

368.txt

X01

Algorytm karuzelowy nie pozwala wywlaszczac procesow systemowych

TAK

NIE

369.txt

X01

Algorytm karuzelowy jest algorytmem z priorytetami dynamicznymi

TAK

NIE

370.txt

X01

Algorytm IRU zastepowania stron pamieci wirtualnej polega na usuwaniu najdluzej zaladowanych stron

TAK

NIE

371.txt

X10

Algorytm IRU zastępowania stron pamięci wirtualnej zapewnia dobre dostosowanie do zestawu roboczego stron

TAK

NIE

372.txt

X01

Algorytm IRU zastępowania stron pamięci wirtualnej zapewnia zawsze najmniejszą liczbę błędów stron (w porównaniu z innymi algorytmami)

TAK

NIE

373.txt

X10

Biblioteki łączone dynamicznie (DLL) ładowane są do pamięci tylko funkcje, do których wystąpi odwołanie

TAK

NIE

374.txt

X01

Biblioteki łączone dynamicznie (DLL) udostępniają wyłącznie funkcje wielowejsciowe (reentrant)

TAK

NIE

375.txt

X01

Biblioteki łączone dynamicznie (DLL) kopiowane są w całości do plików wykonywalnych, zawierających programy z nich korzystające

TAK

NIE

376.txt

X01

Buforowanie operacji dyskowych Eliminuje fragmentację wewnętrzną w systemie plików

TAK

NIE

377.txt

X10

Buforowanie operacji dyskowych umożliwia zapis danych do pliku porcjami różnej wielkości

TAK

NIE

378.txt

X01

Buforowanie operacji dyskowych umożliwia zrownoleglenie procesów

TAK

NIE

379.txt

X01

Dowiązania symboliczne w linuxie zawierają informacje o położeniu pliku na dysku

TAK

NIE

380.txt

X01

Dowiązania symboliczne w linuxie muszą mieć rozszerzenie .ink

TAK

NIE

381.txt

X10

Dowiązania symboliczne w linuxie są to pliki specjalne, które wskazują przy pomocy ścieżek na inne obiekty

TAK

NIE

382.txt

X10

Dynamiczne ładowanie programu zmniejsza wielkość pliku wykonywalnego (exe)

TAK

NIE

383.txt

X01

Dynamiczne ładowanie programu wykorzystuje biblioteki statyczne

TAK

NIE

384.txt

X01

Dynamiczne ładowanie programu polega na tworzeniu połączeń dynamicznych pomiędzy procesami

TAK

NIE

385.txt

X10

Dynamiczne ładowanie programu polega na ładowaniu nakładek programu, gdy są potrzebne

TAK

NIE

386.txt

X01

Dynamiczne ładowanie programu polega na wykorzystaniu bibliotek dołączanych w trakcie wykonywania programu

TAK

NIE

387.txt

X01

Dynamiczne ładowanie programu zmniejsza czas od wydania komendy do uruchomienia programu

TAK

NIE

388.txt

X01

Funkcja systemowa signal tworzy nowy proces

TAK

NIE

389.txt

X01

Funkcja systemowa signal wysyła sygnał do procesu lub grupy procesów

TAK

NIE

390.txt

X10

Funkcja systemowa signal wysyła komunikat do standardowego strumienia błędów

TAK

NIE

391.txt

X10

Funkcja systemowa fork w unice (Linuxie) tworzy nowy proces, realizujący ten sam program co proces-rodzic

TAK

NIE

392.txt

X01

Funkcja systemowa fork w unicie (Linuxie) tworzy nowy watek, realizujacy funkcje podana w argumente

TAK

NIE

393.txt

X01

Funkcja systemowa fork w unicie (Linuxie) tworzy nowy proces, realizujacy program podany w argumente do tej funkcji

TAK

NIE

394.txt

X10

Funkcja systemowa execve w unixie (Linuxie) laduje nowy program w miejsce obrazu pamieci procesu, ktory wywolal te funkcje

TAK

NIE

395.txt

X01

Funkcja systemowa execve w unixie (Linuxie) tworzy nowy proces, realizujacy program podany w argumente do tej funkcji

TAK

NIE

396.txt

X01

Funkcja systemowa execve w unixie (Linuxie) tworzy nowy watek, realizujacy funkcje podana w argumente

TAK

NIE

397.txt

X01

Kanal komunikacyjny z zerowym buforowaniem umozliwia buforowanie operacji dyskowych

TAK

NIE

398.txt

X10

Kanal komunikacyjny z zerowym buforowaniem to kanal, w ktorym aktualnie nie ma zadnych komunikatow

TAK

NIE

399.txt

X01

Kanal komunikacyjny z zerowym buforowaniem może powodować uspienie procesu czytającego z kanału

TAK

NIE

400.txt

X01

Kontekst procesu jest to obraz pamięci, rejestrów i zmiennych systemowych procesu w pewnej chwili

TAK

NIE

401.txt

X01

Kontekst procesu obejmuje informacje konieczne do zablokowania procesu

TAK

NIE

402.txt

X10

Kontekst procesu opisuje stan wszystkich zasobów, z których może proces korzystać

TAK

NIE

403.txt

X01

Kanal komunikacyjny z nieograniczonym buforowaniem może powodować uspienie procesu piszącego do kanału

TAK

NIE

404.txt

X01

Kanal komunikacyjny z nieograniczonym buforowaniem może powodować uspienie procesu czytającego z kanału

TAK

NIE

405.txt

X10

Kanal komunikacyjny z nieograniczonym buforowaniem umożliwia buforowanie operacji dyskowych

TAK

NIE

406.txt

X10

Ogole semaforey dijkstry przyjmuja tylko wartosci 0 lub 1

TAK

NIE

407.txt

X01

Ogole semaforey dijkstry moga blokowac proces przy wykonaniu operacji V

TAK

NIE

408.txt

X10

Ogole semaforey dijkstry moga blokowac proces przy wykonywaniu operacji p

TAK

NIE

409.txt

X01

Ochrona pamieci w systemach operacyjnych okresla prawa dostepu procesow do pamieci dyskowej

TAK

NIE

410.txt

X01

Ochrona pamieci w systemach operacyjnych zapewnia, ze procesy nie zostana zmodyfikowane w trakcie kopiowania do obszaru wymiany (swapping)

TAK

NIE

411.txt

X10

Ochrona pamieci w systemach operacyjnych uniemozliwia dostep procesu do niezaalokowanej pamieci

TAK

NIE

412.txt

X01

Przetwarzanie posrednie to technika przydzielania dodatkowych buforow procesom uprzywilejowanym

TAK

NIE

413.txt

X01

Przetwarzanie pośrednie eliminuje fragmentację zewnętrzną dysku przez przesuwanie plików w tle

TAK

NIE

414.txt

X10

Przetwarzanie pośrednie polega na kopiowaniu danych między dyskiem a wolnymi urządzeniami zewnętrznymi w celu poprawy wykorzystania procesora

TAK

NIE

415.txt

X01

Przetwarzanie potokowe dostępne jest tylko w systemach wieloprocesorowych

TAK

NIE

416.txt

X01

Przetwarzanie potokowe w powłoce sh można je wywołać wpisując dwie komendy w nawiasach okrągłych rozdzielone przecinkami

TAK

NIE

417.txt

X10

Przetwarzanie potokowe realizowane jest przez przekierowanie standardowego wyjścia komendy pierwszej do standardowego wejścia komendy drugiej

TAK

NIE

418.txt

X01

Przydział indeksowy miejsca na dysku charakteryzuje się fragmentacją zewnętrzną przestrzeni dysku

TAK

NIE

419.txt

X10

Przydział indeksowy miejsca na dysku wymaga zapisu tablicy indeksowej na dysku

TAK

NIE

420.txt

X01

Przydział indeksowy miejsca na dysku opisuje położenie pliku jako ciąg segmentów różnej wielkości

TAK

NIE

421.txt

X10

Proces może przejść do stanu zablokowania bezpośrednio ze stanu aktywności

TAK

NIE

422.txt

X01

Proces może przejść do stanu zablokowania w wyniku wykonania funkcji exit (w linuxie)

TAK

NIE

423.txt

X01

Proces może przejść do stanu zablokowania po wyczerpaniu kwantu czasu aktywności

TAK

NIE

424.txt

X01

Proces uśpiony (zablokowany) zawsze usuwany jest z pamięci operacyjnej (kopiowany do obszaru wymiany)

TAK

NIE

425.txt

X10

Proces uśpiony (zablokowany) oczekuje aż inny proces go zbudzi

TAK

NIE

426.txt

X01

Proces uśpiony (zablokowany) oczekuje na zwolnienie procesora przez procesor aktywny

TAK

NIE

427.txt

X01

Powłoka systemu operacyjnego określa zestaw dostępnych języków programowania

TAK

NIE

428.txt

X01

Powłoka systemu operacyjnego określa dostępny zestaw komend wewnętrznych

TAK

NIE

429.txt

X01

Powłoka systemu operacyjnego pozwala definiować zmienne środowiskowe, przekazywane do uruchamianych programów

TAK

NIE

430.txt

X01

Pozycje katalogowe w systemie FAT pozwalają definiować nazwy plików dowolnej wielkości

TAK

NIE

431.txt

X10

Pozycje katalogowe w systemie FAT zawierają informacje o położeniu pliku na dysku

TAK

NIE

432.txt

X01

Pozycje katalogowe w systemie FAT zawierają sumę kontrolną pliku

TAK

NIE

433.txt

X01

Przydział ciągłych miejsc na dysku wymaga zapisu tablicy indeksowanej na dysku

TAK

NIE

434.txt

X01

Przydział ciągły miejsca na dysku jest algorytmem optymalnym, nie dającym się zaimplementować w praktyce

TAK

NIE

435.txt

X10

Przydział ciągły miejsca na dysku charakteryzuje się fragmentacją zewnętrzną przestrzeni dysku

TAK

NIE

436.txt

X01

Planowanie zadań (scheduling) ma na celu synchronizowanie dostępu do zasobów

TAK

NIE

437.txt

X10

Planowanie zadań (scheduling) ma na celu wybór procesu do aktywacji

TAK

NIE

438.txt

X01

Planowanie zadań (scheduling) ma na celu unikanie blokad

TAK

NIE

439.txt

X10

Planowanie średnioterminowe określa kolejność kopiowania procesów do pamięci z obszaru wymiany

TAK

NIE

440.txt

X10

Planowanie średnioterminowe wybiera proces do uruchomienia spośród procesów gotowych

TAK

NIE

441.txt

X01

Planowanie srednioterminowe okresla procesy, ktore nalezy odblokowac

TAK

NIE

442.txt

X01

Sekcja krytyczna to technika synchronizacji dostepu do pamieci wspoldzielonej przez procesor

TAK

NIE

443.txt

X01

Sekcja krytyczna wykorzystywana jest do negocjacji dostepu do semaforow

TAK

NIE

444.txt

X10

Sekcja krytyczna wykorzystywana jest od ochrony danych przed rownoczesna ich modyfikacja przez procesy wspolbiezne

TAK

NIE

445.txt

X01

Sygnały w systemie linux generowane sa wyłacznie przez sterowniki urzadzen zewnetrznych

TAK

NIE

446.txt

X01

Sygnały w systemie linux nie moga byc ignorowane przez procesy

TAK

NIE

447.txt

X10

Sygnały w systemie linux wysylane sa do pojedynczych procesow, grup procesow, procesow okreslonego uzytkownika lub do wszystkich procesow

TAK

NIE

448.txt

X01

Swobodny (zrandomizowany), dostęp do pliku polega na współdzieleniu otwartych plików przez wiele procesów

TAK

NIE

449.txt

X01

Swobodny (zrandomizowany), dostęp do pliku szereguje dostęp wielu procesów do pliku

TAK

NIE

450.txt

X10

Swobodny (zrandomizowany), dostęp do pliku pozwala na występowanie plików nieciągłych, z niezaalokowanymi dziurami

TAK

NIE

451.txt

X01

Stan procesu przechowywany jest w jądrze systemu operacyjnego w obszarze uarea

TAK

NIE

452.txt

X01

Stan procesu Identyfikuje procesy, które nie wymagają dostępu do urządzeń zewnętrznych

TAK

NIE

453.txt

X10

Stan procesu określa, czy procesor jest przydzielony do procesu

TAK

NIE

454.txt

X10

Stronicowanie na zadanie wymaga tworzenia obrazu pamięci w obszarze wymiany

TAK

NIE

455.txt

X01

Stronicowanie na zadanie wykorzystuje segmentację pamięci

TAK
NIE

456.txt

X01

Stronicowanie na zadanie włączane jest na zadanie procesu (przez wywołanie funkcji swapon)

TAK
NIE

457.txt

X10

Ścieżka do pliku to ścieżka bezwzględna określająca położenie pliku względem katalogu głównego

TAK
NIE

458.txt

X01

Ścieżka do pliku może być względna, bezwzględna lub symboliczna

TAK
NIE

459.txt

X01

Ścieżka do pliku to ścieżka symboliczna określająca położenie pliku względem katalogu domowego użytkownika

TAK
NIE

460.txt

X01

Warunek postępu w synchronizacji między procesami to zapewnienie, że dwa procesy nie znajdują się w tym samym czasie w sekcji krytycznej

TAK
NIE

461.txt

X10

Warunek postępu w synchronizacji między procesami to wymaganie, aby proces nie był usypiany przy wejściu do sekcji krytycznej, gdy nie ma w niej innych procesów

TAK
NIE

462.txt

X01

Warunek postępu w synchronizacji między procesami zapewnia, że proces nie jest wywłaszczany, gdy znajduje się w sekcji krytycznej

TAK

NIE

463.txt

X01

Warunek wykluczania w synchronizacji między procesami zapewnia, że proces nie jest wywłaszczany, gdy znajduje się w sekcji krytycznej

TAK

NIE