**Systemy operacyjne 1 09.12.2023 1 czesc**0:00w wykładzie kursu Wielodostępne Systemy Operacyjne I.0:05Plan na dzisiaj, najpierw omówienie prac domowych, rozwiązań,0:10które Państwo nadesłali, a następnie kolejny materiał dotyczący0:15zarządzania procesami.0:21Popatrzmy na zadania domowe.0:23To jest pierwsze zadanie dotyczące informacji w Iwęźle.0:30Ja przełączę w tym momencie slajd, którego należało korzystać0:39przy rozwiązywaniu tego zadania.0:51Jak wspomniałem, zadanie dotyczy informacji, które znajduje się w Iwęźle0:58po wykonaniu określonego zadania.1:00Zadanie dotyczy określonego zestawu instrukcji.1:03Tutaj był ten zestaw instrukcji, a teraz popatrzmy na przykłady Państwa odpowiedzi.1:09Ja tutaj zestawiłem wybrane odpowiedzi.1:15Tutaj są niektóre teksty, bo Państwo przesyłali czasami mailami1:18w formie oddzielnych plików, a czasami w treści maila była odpowiedź.1:24Popatrzmy na przykłady odpowiedzi.1:30A tutaj widzimy, zaczęło się od omówienia.1:40To jest przykład takiej niepełnej odpowiedzi.1:46Tutaj jest informacja, jak sprawdzający treść zadania rozumiał to,1:54co się działo w systemie.1:56No a następnie próba odpowiedzi.2:04Popatrzmy.2:06Prawidłowa odpowiedź to powinna po prostu zawierać informacje dotyczące2:11wszystkich tych elementów, które są zawarte w Iwęźle.2:16No a tutaj widzimy, nie wszystkie z nich są tutaj zamieszczone.2:22Niektóre są, rozmiar jest dobrze, liczba dowiązań.2:26Trzy.2:26Czas ostatniej modyfikacji w porządku, czas ostatniego dostępu również.2:35No ale nie ma szeregu, widzą Państwo, których informacji.2:41Jaki to jest typ pliku, jakie są prawa.2:43No prawa można było sobie co prawda samemu podać, bo w treści zadania ich nie ma.2:48Ale pełna odpowiedź powinna zawierać wszystkie te elementy tutaj wymienione.2:55Popatrzmy.2:57Ja na razie to pierwsze zadanie będę omawiał z Państwem.3:03Tu jest przykład dobrej odpowiedzi z wyjątkiem jednej informacji.3:12I to jest dobre, to jest zwykły plik.3:16Natomiast nie należało, ta informacja, tu na jedno chcę zwrócić uwagę.3:22System UNIX nie zawiera...3:26Zawiera i węźle informacji, jaki to jest rodzaj tego pliku zwyczajnego.3:34No ten plik zwyczajny to może być tekstowy, może być przecież muzyka, może być grafika i inne informacje.3:41To jest po prostu ciąg bajtów, także należy o tym pamiętać.3:44No w tym przypadku jest to plik tekstowy, jest to dobra, jest to plik zwykły i to jest dobra odpowiedź,3:52bo mamy przykłady, które Państwu podawałem.3:55Podawałem plik zwykły.3:56Plik typu katalog, plik typu dowiązanie symboliczne i tak dalej, plik specjalny i popatrzmy na prawidłową odpowiedź.4:06No tutaj pewne domyślne, ponieważ nie były w treści zadania, tak zostało.4:12Liczba dowiązań trzy.4:15Gdy nowy plik powstaje, to pamiętajmy, no to będzie jedno dowiązanie, znaczy gdy kolejne nazwy pliku,4:24bo plik jako obiekt,4:26to jest jeden.4:28Natomiast on może być kilka nazw.4:30Więc rozróżniajmy pojęcie plik jako obiekt, który jest zapisany na dysku i ma swój i węzeł, jeden, a nazwy, nazw może być wiele.4:41I właśnie w przypadku pewnej liczby nazw, mamy pewną liczbę dowiązań.4:47Identyfikator właściciela, identyfikator grupy, to również nie było w treści zadania, więc można było sobie wpisać, co kto uważał.4:56Rozmiar, tu jest podane dlaczego, bo mamy dwukrotne wykonanie polecenia zapisu.5:02I drugi zapis dotyczył uzupełnienia tego, co było.5:08Czas modyfikacji, to jest w tym wypadku, to będzie ostatni.5:13Modyfikacja pliku, tam było 21.30.5:16Może teraz sobie też przesunę tutaj treść zadania, żeby Państwo lepiej to widzieli.5:23Widzimy, to jest 21.30.5:25Było dopisanie drugiej informacji generowanej przez aplikację Date.5:40Czas ostatniego dostępu, to jest 22.30.5:47Wtedy był odczyt pliku, prawda?5:49I to jest tu zapisane.5:51A czas ostatniej zmiany informacji jest również 22.30.5:55To chcę podkreślić, ponieważ informacja o dostępie do zawartości pliku na dysku jest również w eventzie.6:07Podkreślam to dlatego, że bardzo w wielu pracach tutaj był błąd.6:13Czyli jeśli był dostęp do informacji pliku, to jeśli to była ostatnia operacja zmieniająca informację w eventzie,6:23no to również ten czas ostatniej zmiany w eventzie powinien być tutaj wpisany.6:33No i następnie są wskaźniki adresowe.6:36W tym przypadku to jest zestaw informacji generowanych przez Date.6:42Aplikacja Date jest na tyle mała, że pewnie w jednym bloku się to zmieści.6:47Teraz popatrzmy na inne odpowiedzi.6:53Tu jest dobra odpowiedź.6:59Tu jest właśnie zła odpowiedź, bo event, jak widzimy, zawiera informacje także dotyczące dostępu do pliku.7:09A tutaj błąd zaznaczyłem na czerwono zamieszczony tutaj.7:20Dobrze.7:22Jeszcze parę tutaj przykładów takich błędów.7:30Popatrzmy, tutaj mam jeszcze.7:46Popatrzmy, co tutaj mamy.7:50W tym przykładzie to jest treść zadania.7:53A, to jest sama treść zadania.7:56Niektóre osoby takie coś mi przysyłały, więc proszę zwrócić na to uwagę.8:02Przekopiowanie treści zadania to nie jest rozwiązanie, prawda?8:07Żartuję trochę, ale rozumiem, że ta osoba sobie zażartowała, która takie coś mi tu wysłała.8:20Typ pliku, plik.8:25To jest zła odpowiedź.8:28Brak konkretnej odpowiedzi.8:32Wiemy, że w tym wypadku to jest plik zwykły.8:36Liczba dowiązań.8:39Nie ma tutaj tej informacji.8:41Plik A, plik B i co dalej.8:44Chodziło o liczbę, a nie nazwy.8:48Bo nazwy są, to jest błąd taki zasadniczy, bo Państwo pamiętają, że nazwy plików są w pliku typu katalog, a nie w i-węźle.9:00Rozmiar pliku 30 bajtów.9:05No, jest 60 powinno być.9:08I tutaj widzimy czas ostatniej zmiany informacji w i-węźle 20.9:15No, ta osoba nie zauważyła, że9:17w i-węźle znajdują się informacje właśnie o ostatniej modyfikacji i ostatnim dostępie, prawda?9:24No i tak dalej.9:25Także tu widzimy przykład błędnej odpowiedzi.9:34Zobaczmy tu jeszcze.9:41Tu niektóre osoby przysyłały mi odpowiedź tego typu, że sobie zrobiły...9:46No, ćwiczenie, bardzo dobrze to ćwiczenie jest,9:50tylko te informacje tutaj generowane w ramach tego ćwiczenia,9:56no nie mają związku z treścią zadania, widzą Państwo.10:01Tu są jakieś czasy dostępu niezwiązane z treścią zadania,10:07więc bardzo proszę na to zwrócić uwagę,10:10że prawidłowa odpowiedź powinna dotyczyć tego zadania, które zostało wydane.10:15Zwracam na to uwagę, ponieważ podobne zadanie prawdopodobnie będzie10:19na tym końcowym sprawdzianie, takim jak egzamin na końcu kursu.10:26Dobrze, przejdziemy teraz do...10:28Chyba, że ze strony Państwa są jakieś pytania.10:31Jeśli nie, to przejdziemy do tego drugiego.10:40No, sprawdziłem.10:45A, no, to znaczy, mogło być tak, że przy tej dużej liczbie,10:51no, mogłem kogoś, że tak powiem, no, pomylić się i nie zauważyć,10:57bo się mógłbym zmieszać z innymi.10:59Bardzo proszę, Pan, prześle mi forward swojego maila11:05i ja oczywiście uzupełnię i też sprawdzę Pana pracę,11:08czy jak będzie dobra, to oczywiście to były zadania punktowane,11:12więc oczywiście Panu odpowiednie punkty,11:14odpowiednie punkty wystawię, także proszę o forward.11:17Natomiast inne osoby...11:24No, obecność jest zaliczona, tylko jest zero punktów wtedy11:31z tego zadania.11:34Zadania to dla osób, które...11:37Dużo osób mi nie przysłało w ogóle odpowiedzi11:39albo, nie wiem, może z jakiegoś powodu ja ich nie dostałem,11:42więc jeśli...11:44Jeśli wysłały mi wcześniej, to bardzo proszę,11:46no, dobrze, że Pan na to zwrócił uwagę,11:48proszę o forward.11:50Natomiast te osoby, które nie wysłały,11:52mogą jeszcze dosłać te odpowiedzi,11:54ja zaliczę obecność,11:57a te rozwiązane zadania są istotne,12:01dlatego że w ten sposób Państwo przygotowują się12:04do tego kończego sprawdzianu podsumowującego cały kurs.12:09Także, no...12:12Tak, żeby mogły te osoby uniknąć tych, no, błędów,12:17na które ja zwracam tutaj uwagę.12:21Ale teraz od strony merytorycznej,12:24któreś sprawy ze strony Państwa tutaj sprawiły pewien problem.12:30No to popatrzmy, jakie informacje są w wiweźle, popatrzmy.12:51Jest w wiweźle czas ostatniego dostępu do pliku?12:55Jest.12:57No to jak on się zmienił,12:59to zmieniła się informacja w wiweźle.13:04Tak?13:09No jeśli którakolwiek zmiana13:13dotycząca tego, co jest w wiweźle nastąpiła13:17w odpowiednim momencie czasu,13:19no to ten czas będzie czasem13:21ostatniej zmiany informacji w wiweźle.13:23No to chyba jest dość...13:26Teraz może ktoś na to nie zwrócił uwagi, ale proszę.13:29Proszę rozumieć treść tego, co jest w wiweźle.13:34Tak? Już Pan złapał?13:37Dziękuję bardzo.13:40Czy może ktoś inny ma jakieś pytanie?13:42Bardzo dobrze, że Państwo teraz...13:44No teraz jest pora, żeby ewentualnie uzupełnić właśnie,13:48czy wyjaśnić sobie pewne sprawy.13:53Przejdziemy do drugiego zadania.13:56Ja tutaj jeszcze dopowiem, podobnie jak kolega,13:57nie mam zaliczonej obecności, ale przysłałem...14:01Tak jak mówię, to dotyczy wszystkich osób.14:04Bo właśnie zaniepokoiło mnie to,14:06że było bardzo dużo nieprzysłanych odpowiedzi.14:11Nie wiem, z jakiego powodu.14:13Ja starałem się wszystkie maile od Państwa14:16zczytać. Może one jakoś nie dotarły do mnie.14:19Nie wiem.14:21Przepraszam, a jaki jest właściwie14:23Pana adres mailowy?14:25Bo też rozmawiałem z kilku osobami14:27i nie byliśmy pewni, czy na pewno odpowiedni adres,14:29na odpowiedni adres wysłaliśmy.14:31Kruz? Kruz?14:33No, gdy ja do Państwa przesyłam,14:36rozsyłam informacje dotyczące14:39zadań domowych, no to wystarczy odpowiedzieć na to.14:44Tam jest kruz.witedu.pl14:51Dostają, ale do Państwa docierają te moje maile,14:54moje maile rozsyłane przez system UBI, czy nie?14:58Tak, tak, docierają.15:07Nie no, ale w nadesłanej odpowiedzi15:10jest przecież mail osoby wysyłającej, prawda?15:14No to wystarczy z niego skorzystać.15:19Ja rozumiem, że Państwo różnych tutaj mogli15:22tych aplikacji pocztowych,15:24skorzystać może któreś tutaj.15:27No nic, w każdym razie bardzo proszę,15:30najprościej jest odpowiedzieć na mail,15:32który ja wysłałem z treścią zadania15:35i wtedy na pewno powinno do mnie trafić.15:40Dobrze, przejdziemy teraz do...15:49Nie, nie, to tytułować dalej,15:51tak jak ja prosiłem.15:52A, bo może były źle zatytułowane,15:54te prace.15:55No ja na pierwszym...16:00No to będzie FORWARD po prostu.16:03Nie, napisać FORWARD,16:05ale FORWARD i że to jest SO1,16:09że jest to wykład, który...16:12że są to zaoczne,16:13Z literkę wpisać, który wykład,16:15no bo ja w sumie dostaję ponad 400 maili,16:19więc łatwo mi, jeśli inaczej są zatytułowane,16:22niż ja prosiłem,16:24no to łatwo je pominąć,16:27bo no dostaję także w innych sprawach maile16:30i ze szkoły i od innych osób zupełnie.16:33Także proszę.16:35Tak, proszę.16:37W tym przypadku te zadania nie były przypisane16:39do konkretnych wykładów,16:40jak w pierwszym i drugim.16:42Czy w takim razie powinniśmy wysyłać na przykład16:44jedno zadanie, wykład trzeci, jedno wykład czwarty?16:46Nie, nie, wygodnie jest,16:48jak nawet w jednym pliku Państwo odpowiedź dadzą.16:52Czyli na przykład...16:53bo ja na przykład zatytułem W3 i 4.16:57No i bardzo dobrze.16:59I co, nie zauważyłem tego od Pana?17:02Nie, nie mam ubiegobecności.17:04No to proszę, bardzo proszę.17:05No nie wiem, bo ja starałem się wszystkie odpowiedzi złapać,17:08które do mnie przyszły.17:11No może coś nie przyszło, więc proszę o forward.17:15Tego, co było wysłane.17:17Bo wtedy ja będę wiedział, że to było wysłane wcześniej.17:20I naliczę, jeśli będzie dobrze, to naliczę punkty.17:23Natomiast...17:24Ja właśnie a propos tego mam jeszcze jedno pytanie,17:27ponieważ w zeszłym miesiącu to był 11 listopada17:30i ja błędnie założyłem, że tak jak wszystkie zdjęcia17:33się przesuwają o tydzień,17:35to że tutaj to możemy też podesłać tydzień temu.17:37No i ja napisałem właśnie w tym mailu,17:39że właśnie przepraszam za tą nieuwagę.17:42No i wysłałem te zadania, no i tam też napisałem,17:44że jeżeli nie dostanę punktów, to jak najbardziej to rozumiem.17:49Ale też nie dostałem odpowiedzi, więc...17:51Nie, to proszę jeszcze raz w ogóle.17:52No nie wiem, nie wiem.17:54Może były jakieś problemy z pocztą.17:57Także bardzo proszę.17:59O, widzę, że to już od pana Zawadzkiego dostałem tutaj.18:03Forward, bardzo dobrze.18:04I bardzo dobrze pan zatytułował.18:06No jest forward, to automatycznie.18:08SO 1, Z 2, 3 i 4.18:10Bardzo dobrze.18:11I nawet wygodnie na przyszłość na dzisiejsze,18:14odpowiedź na dzisiejsze zadania,18:16również w formie jednego pliku,18:20w którym będzie odpowiedź.18:21Obie odpowiedzi.18:22Bo mi łatwiej sprawdzać coś będzie po prostu.18:25Dobrze, może przejdziemy teraz do tego drugiego zadania.18:31Ja to zamknę.18:33O, zaraz.18:37Nie, anuluj.18:41A nie, to już nie będzie potrzebne.18:46O, to się przyda, sposób przechowywania,18:51bo tego dotyczyło właśnie kolejne zadanie.19:00Tutaj popatrzmy z kolei na wykłady odpowiedzi.19:09Popatrzmy.19:13Rozwiązując to zadanie należałoby się kierować po prostu tym,19:17tą informacją, którą ja podałem na wykładzie,19:20dotyczącą tych reguł,19:23dotyczących właśnie przydziału miejsca na dysku.19:27W treści zadania mieliśmy tak.19:31Rozmiar pliku, który trzeba zaadresować.19:35W uwzględnieniu tam był podany także rozmiar bloku 16 KB,19:41rozmiar fragmentu 4 KB i była podana mapa bitowa.19:47No i żeby zmieścić całą informację w pliku,19:50no to trzeba, widzimy Państwo,19:52użyć tutaj trzy bloki i trzy fragmenty.19:55No i zasada działania jest taka,19:57że system operacyjny przeszukuje mapę bitową20:02i najpierw znajduje całe pełne bloki20:08i znajduje, tutaj widzimy Państwo,20:10widzą Państwo te bloki, które zostały na początku,20:14a reszta, i to mi się podobało w tym zadaniu, w tej odpowiedzi,20:18a następnie, patrzmy, to proszę się kierować20:22tego typu zadaniem, żeby je rozwiązać,20:25tymi regułami, które tutaj są podane po prostu.20:29A następnie szuka miejsca na końcówkę,20:34tą, która zawiera trzy fragmenty.20:38Istotne jest, zgodnie tutaj z regułami,20:41że to mają być trzy fragmenty w jednym bloku i obok siebie.20:45I widzimy tutaj, że jedyne takie miejsce,20:48to jest tutaj, prawda?20:50Tu są w jednym bloku trzy wolne fragmenty,20:55ale są przedzielone, to nie, to źle.20:58Więc końcowa mapa bitowa powinna wyglądać tak, jak tutaj.21:03Ta końcówka pojawi się w tym drugim bloku.21:09Druga część zadania dotyczyła informacji w iweźle21:14i tutaj tej drugiej części odpowiedzi nie ma.21:18Ale tutaj popatrzmy na kolejne zadanie.21:22Tutaj jest ta odpowiedź.21:24Bo co w iweźle, jakie informacje adresowe się znajdują?21:31I to, jeśli ktoś nie zapisał poprzednio, to proszę.21:34Informacje adresowe w iweźle są w takiej kolejności,21:38to jest ważne, w jakiej jest zapisywana informacja pliku.21:43Czyli tak, jeśli początek pliku to będzie od adresu 2,21:4812 w tym bloku, no to na pierwszym miejscu w iweźle21:52znajdzie się 12.21:54Jeśli dalej od 16, to będzie 16.21:57A końcówka zawsze jest na końcu.22:02Końcówka informacji pliku i adres, od którego się zaczyna.22:08Proszę.22:11No fragmenty.22:14Adres jest początkowego fragmentu.22:18To jest, a to jest, bardzo proszę coś zapisać.22:21Nie ma, nie będą trzy adresy.22:23Nie.22:24Będzie tylko jeden początkowy.22:29Jeśli ktoś poprzednio tego nie zapisał,22:31to proszę to zapisać.22:32Jest tylko jeden adres początkowy.22:35No bo tutaj o tym, w tych regułach tego nie ma, prawda?22:39Ale ja to mówiłem poprzednio.22:41Jeśli ktoś, no nie zauważył, nie zwrócił uwagi,22:44to proszę na ten środek.22:46Znaczy ja mówiłem o tym kilka razy.22:48No popatrzmy teraz na przykłady błędów.22:52Pewnie gdzieś tu będą.22:54A tu było, to jeszcze.22:59No mam tutaj.23:18Ja mogę, ale no, ewentualnie w przerwie lub po wykładzie mogę Państwu pokazać.23:37No niestety tych odpowiedzi dobrych było bardzo mało tym razem.23:41No tu popatrzmy, na czym błąd polegał.23:44Tam były potrzebne trzy fragmenty.23:48A tu jeszcze odpowiadający napisał,23:52że jeszcze zostanie zarezerwowane miejsce o tutaj23:55i będzie adres tutaj.23:59Ja to na czerwono zaznaczyłem o to24:02i to jest błąd, prawda?24:04Dotąd byłoby dobrze.24:06No po co mu dwa dodatkowe fragmenty?24:08No być może to była jakaś taka pomyłka,24:11no mechaniczna bym powiedział, a nie logiczna.24:16Ale no proszę starać się takich rzeczy unikać.24:20To popatrzmy może jeszcze inny przykład.24:31Nie, dotyczące adresowania było sporo dobrych.24:36A tu jest taki, gdzie mamy dwa błędy.24:39Zobaczmy, co tutaj było w tym zadaniu.24:46O nie, tu nie, ten pik mi się wyświetlił niestety.25:13Coś mi się nie chce powtarzać.25:15Co innego proszę, a co innego mi wyrzuca tutaj.25:25No nic, a w każdym razie bardzo proszę,25:28jeśli są ze strony Państwa jakieś pytania,25:30to chętnie spróbuję odpowiedzieć.25:33W tym adresowaniu, no to Państwo proszę się kierować25:40tym, co mówiłem.25:42Czyli tą dobrą odpowiedzią, którą pokazywałem.25:44I były właśnie, były błędne odpowiedzi takie,25:49że na przykład ktoś wykorzystał te trzy fragmenty,25:52no które są przedzielone, no to jest źle.25:54Czy też odpowiedź na przykład taka,25:58które były wykorzystane o te dwa26:00i jeden z następnego bloku.26:02No to też jest zła odpowiedź.26:04Czyli ta końcówka musi być w jednym bloku26:06i to muszą być kolejne fragmenty.26:09Bo od tego początkowego adresu,26:12gdy zacznie czytać,26:19no to czyta po kolei tyle, ile wynika z rozmiaru pliku.26:24A rozmiar pliku ma w imięźle, prawda?26:27No korzysta z informacji w imięźle.26:29I dotyczących adresów, i dotyczących tego,26:32co należy odczytać.26:34Czy ze strony Państwa są jakieś pytania26:36dotyczące tego zagadnienia?26:42No wynikająca z tego, że tak,26:50plik ma pewien rozmiar, prawda?26:52I w tych ostatnich, w tej końcówce fragmentów26:57nie wszystkie, no były wykorzystane, prawda?27:01W ostatnim fragmencie był część.27:03A ja może to, mam chyba nawet taki u siebie.27:12Tak, dokładnie.27:14I to, no to co było niewykorzystane,27:16no to jest to, co możemy uznać właśnie27:19jako stratę wynikającą,27:21ale z wewnętrznej fragmentacji pamięci operacyjnej.27:28Chcę na chwilkę spojrzeć.27:30A, chciałem tu jedną rzecz może jeszcze Państwu pokazać.27:34Wrócę tutaj do systemów plików.27:42Dobrze, że Pan o tym wspomniał,27:44bo ja nawet sobie przygotowałem odpowiedź na Pana pytanie.27:56Popatrzmy tutaj, jak to wygląda, o.28:01Jak wygląda teraz informacja tego pliku?28:04Jak czyta tą informację system operacyjny?28:09Zaczyna od, jeśli tu adres pierwszy jest 12,28:12potem czyta dalej.28:15Kolejny adres to jest 16,28:17tu widzimy będzie 16.28:19Kolejny, te bloki są rozrzucone na dysku, prawda?28:23Ale on traktuje to jako ciąg informacji w tym pliku.28:26Ja mówię, jak wygląda ta informacja w pliku,28:28która w rezultacie mieści się w różnych blokach, prawda?28:32A końcówka jest od adresu 5.28:35I tu widzimy oto białe,28:38to jest właśnie to, co z tego ostatniego fragmentu28:42jest niewykorzystane.28:48Także tu specjalnie przygotowałem ten dodatkowy plik,28:50żeby Państwu lepiej jeszcze to pokazać.28:53Czyli tak, na dysku informacje są rozrzucone,28:56no ale jeśli czytamy plik, no to to jest ciąg informacji.29:01Pamiętają Państwo, to jeszcze podkreślam.29:04System UNIX nie zawiera żadnych informacji29:07o formacie tej informacji zawartej w pliku.29:12To jest dla niego ciąg bajtów.29:15Natomiast oczywiście różne aplikacje29:17mogą wprowadzać swoje formaty29:20i odpowiednia aplikacja musi być użyta,29:23żeby ten format odczytać, prawda?29:28Czy jeszcze jakieś pytania z tym związane?29:30Bardzo proszę, no mogą być.29:35Było zadanie dodatkowe, a ono nie było punktowane.29:39No zakładam, że ono było na tyle,29:40no niewiele osób mi na nie odpowiedziało,29:43ale no wrócimy tutaj, popatrzmy.29:48W takim razie ja to pozamykam ten dodatkowy plik,29:54bo już za dużo się ich tutaj robiło.30:10A, bo to był już, część trzecia była, tak?30:17Muszę wrócić do części drugiej.30:40Są dostępne materiały.30:44Część druga była tutaj.30:47Muszę ją troszkę zamknąć, niestety.31:04Tu widzą Państwo ten slajd dotyczący...31:08No jeśli...31:09Mamy tak.31:11Na co...31:13Na czym polegają typowe błędy?31:17Ile adresów mieści się w jednym bloku?31:24Pamiętajmy i to proszę sobie zapisać,31:26zaznaczyć, że to, to jest liczba adresów,31:29które się mieści w jednym bloku.31:32Czyli jeśli był...31:35W treści zadania jest rozmiar bloku podany31:38i rozmiar adresu podany,31:41to tutaj był adres, który miał rozmiar 4 bajty,31:45czyli 32 bitów, prawda?31:48No to tych mieści się w 4 kilobajtach 1024.31:53W treści zadania było 16 kilobajtów,31:56a chyba adresowanie było 64-bitowe.32:00No to 16 kilobajtów blok,32:05a adres 8 bajtów.32:07No to 2048 będzie tutaj, prawda?32:12No a dalej analogicznie liczymy to wszystko.32:15Czy w miarę jasne to jest?32:19No 64 bity to jest 8 bajtów, tak?32:29A tu było 16 kilobajtów blok,32:34to ile się mieści?32:36Ile to 8 bitów razy ile będzie 16 kilobajtów?32:43No 2048 dokładnie.32:46Także tu należało po prostu w miejsce 40,32:52no tutaj będzie inaczej,32:54nie 48 tylko 12 razy 16, tak?32:57A tutaj będzie 2048 razy 16,33:04tu do kwadratu razy 16, no i tak dalej.33:09Po prostu te same reguły trzeba zastosować.33:11Także ja uznałem, że to zadanie jest na tyle proste,33:14że nie ma co.33:16No ale niestety błędy mogą być.33:19Czy jeszcze jakieś inne pytania są?33:21Bo materiału mamy bardzo dużo33:23i ja muszę już szybko przeskoczyć do nowego materiału.33:28Jeśli nie, to ja pozamykam te rzeczy,33:32które przechodzimy do zagadnień zarządzania procesami,33:45bo na tym skończyliśmy poprzedni wykład.33:48Czyli ja poprzednio omówiłem samo pojęcie procesu,33:51tworzenie procesu, sygnały33:53i zagadnienie kończenia wykonywania procesu.33:57I od tego momentu będę chciał teraz dalej.34:02Mówić.34:04Tu wspominałem o tym, o sygnałach, prawda?34:07Że tu był podany na tym ostatnim wykładzie34:10na samym końcu składnia polecenia KIL34:14i różne typy sygnałów, które można wysyłać do procesu.34:17Popatrzmy dalej.34:19Co się dzieje, gdy proces się kończy?34:24No może się skończyć w sposób naturalny,34:27ale może się skończyć także, że no coś się zawiesiło,34:30sygnał jakiś wystąpił.34:32I nie można go dokończyć.34:34W każdym razie w obu wypadkach34:37wywoływana jest funkcja systemowa EXIT34:42i proces przechodzi w taki stan zombie.34:46On zwalnia swoje zasoby,34:48czyli zwalnia ten blok kontrolny procesu,34:55oczyszcza kontekst,34:57czyli z pamięci operacyjnej zostanie to wyrzucone,35:00ale ciągle jego pozycja, ten PID, proces ID,35:05ciągle znajduje się w tablicy procesu.35:08Istotne jest to, że proces macierzysty,35:11który wywołał potomka,35:13ciągle czeka na informację o jego zamknięciu.35:17A ta informacja wysyłana jest przez system operacyjny.35:21A on w tym czasie wykonuje funkcję systemową WAIT,35:24czyli jest uśpiony,35:26czyli przykładowo login shell jest uśpiony w tym momencie.35:30Natomiast co pewien czas35:32jądro systemu operacyjnego przeszukuje informację,35:37czy któreś procesy,35:40takiego procesu, który w danym momencie jest w systemie,35:44nie jest w stanie zombie.35:46Jeśli znajdzie, to informuje proces macierzysty o tym,35:51że potomek o takim PID się zakończył35:56i przekazuje parametr funkcji EXIT.36:00Może być to naturalne zakończenie procesu,36:03to wszystko jest w porządku,36:05ale gdyby na przykład to zakończenie procesu36:07wystąpiło z powodu jakiegoś błędu,36:09to będzie kod błędu wtedy tam zawarty.36:12I to dostarcza do procesu macierzystego.36:15Także przykładowo, gdy są jakieś błędy,36:18to shell może w pewnych sytuacjach36:21informację o takim błędzie wyświetlić.36:24Czyli proces macierzysty uzyskuje informacje36:28od systemu operacyjnego,36:30który z możliwych potomków jego skończył pracę,36:33no i wtedy się budzi i dalej działa.36:37Wtedy dopiero, gdy ta informacja zostanie przekazana36:41do procesu macierzystego,36:44jądro systemu operacyjnego dopiero zwalnia36:48tą pozycję, to miejsce w tablicy procesu.36:51To jest druga struktura istotna.36:54Tam są wszystkie, wymienione są wszystkie procesy,36:57które w danym momencie działają,36:59czyli gdy nowy proces powstaje,37:01mówiłem o tym, to dostaje miejsce w tablicy procesu,37:05no w tym momencie, gdy się kończy,37:07ale to dopiero wtedy, gdy system operacyjny37:10przekaże to do potomka, do tego, do macierzystego.37:15No tu, ale to już taka informacja dodatkowa,37:18a co by się stało, gdyby macierzysty zginął37:22i go nie ma?37:24No to wtedy taki proces w tym stanie zombie37:26będzie tak sobie, aż do, że tak powiem,37:31no będzie ciągle w systemie, niestety.37:34W tej tablicy procesu.37:37Także o tym trzeba pamiętać, że...37:40No to tak jak podaję, tak jak to pamiętam ciekawostkę,37:44były takie przypadki, że w jakimś takim,37:46no w systemie IBM kiedyś tam jeszcze,37:49w tamtych czasach, no zdarzały się takie sytuacje,37:52że po roku na przykład ileś procesów zombie było,37:56i zajmowało miejsce w tablicy procesów,37:59no a tablica ma liczbę procesów skończoną,38:03więc jeśli dużo będzie w stanie zombie,38:06zajmowało miejsce w tablicy,38:09to liczba nowych procesów będzie ograniczona,38:12no zmniejszona po prostu.38:15Na to trzeba zwrócić uwagę.38:17No to jako ciekawostkę tylko podaję.38:19Możemy podglądać, czyli sprawdzać informacje38:23o procesach, które są w systemie.38:25Do tego celu służy polecenie proces status.38:28Takie typowe opcje, które można zalecić do wykorzystania,38:34to są opcje E i F, która podaje szereg informacji o procesach.38:40No takie trochę szersze niż samo polecenie PS,38:45bo ono również pewne informacje o procesach wysiewa.38:48I to popatrzmy dokładniej.38:50Składnia to jest PS i opcje tego polecenia.38:54No bez opcji wyświetla taką skróconą informację,38:59a natomiast te opcje pozwalają szereg39:02dodatkowych informacji uzyskać.39:04Na przykład wszystkie procesy w systemie39:09mogą być wyświetlone, no jeśli użytkownik39:11będzie miał takie prawo oczywiście.39:13F to taka pełniejsza informacja o procesach.39:17Można uzyskać, no to dla administratora.39:20Jeśli poda się identyfikator terminali,39:22to będzie informacja o procesach,39:24z tego terminala, z tym terminalem związanych.39:28Jeśli poda się opcję U i identyfikator użytkownika,39:32to administrator może uzyskać informację,39:35jakie procesy dany użytkownik ma jako właściciel.39:40A tu podaje przykład informacji polecenia PS bez opcji.39:47One są wtedy wymieniane, te procesy,39:52tylko te, które są związane z terminalem,39:55na którym użytkownik pracuje.39:57No i tu widzimy, no ja używałem tego w systemie40:01PAKARD-a, tam takim wygodnym login shell-em40:05był shell KORNA, no to mamy informację,40:09proszę popatrzeć, identyfikator tego procesu,40:12shella KORNA, który był procesem obsługującym sesję40:17i widzimy dla polecenia PS został uruchomiony proces40:22w identyfikatorze 428. Widzimy, tu możemy właśnie,40:26dzięki poleceniu PS, możemy Państwu zobaczyć,40:29jaki jest identyfikator tego terminala,40:33na którym dana osoba pracuje, no i czas.40:37Także tutaj to widzimy, natomiast zalecana jest opcja EF,40:41jeśli ktoś chce uzyskać więcej informacji.40:45Tu na jedną rzecz chcę zwrócić uwagę.40:48Mamy identyfikator właściciela,40:52danego procesu.40:54Mamy identyfikator procesu PID,40:59identyfikator procesu macierzystego,41:02to jest odstęp parent process ID,41:05no i odpowiednio wykorzystanie procesora,41:09czas uruchomienia procesu,41:13terminal, identyfikator terminala,41:16z którym związany jest ten proces, czas wykonywania41:21i polecenie, które spowodowało dany proces.41:28Tu widzimy znowu Shell Corna, który obsługiwał tą sesję.41:35Tutaj byłem jako AS1, taką nazwę przyjąłem jako użytkownik,41:41tu widzimy identyfikator.41:43Ale co jest ciekawe, tu widzą Państwo,41:46że procesem macierzystym dla tego Shell Corna41:49obsługującym sesję,41:51jest proces o identyfikatorze 1.41:54A ten 1, to jest jaki proces?41:58On się nazywa init.42:00Czyli init jest tym procesem,42:03który jest macierzysty dla procesów Shell'a42:10obsługujących sesję użytkownika.42:13Tu jeszcze pokazuję taki przykład,42:16taki proces Get TTY,42:19którego procesem macierzystym jest też ten init,42:25a który obsługuje uruchomienie i pracę terminala.42:32W tym wypadku on dotyczy jakiegoś terminala TTY0P5.42:39Init tak naprawdę, ale o tym będę mówił później,42:46uruchamia takie procesy Get TTY,42:48dla wszystkich możliwych terminali.42:53No i zachęcam w trakcie pracy do,42:56że tak powiem, trenowania tego polecenia proces status,43:01dla sprawdzania jakie tam procesy działają w systemie.43:05A w szczególności jakie procesy dotyczą użytkownika,43:09jak to wyglądają i te informacje o procesach43:13w danej chwili działających.43:16Teraz popatrzmy jak wykonywane są polecenia.43:30Shell, pamiętamy, że jest to program, aplikacja,43:36no i odpowiednio ona działa jako pewien proces43:40i43:42ona, jeśli popatrzymy na Shell, który obsługuje sesję,43:50interpretuje polecenia wydawane przez użytkownika,43:53to on bierze pod uwagę trzy typy poleceń.44:00To może być klik wykonywalny,44:04który zawiera kod skompilowanego programu,44:09to wtedy on uruchomi odpowiedni proces,44:10jako proces macierzysty.44:12Ale to może być też plik wykonywalny,44:15który zawiera ciąg poleceń dla Shella,44:18tak je nazywamy skryptem,44:21no i tam mogą być polecenia, które wymagają utworzenia nowego procesu,44:28ale mogą być takie polecenia obsługiwane przez sam Shell.44:32Takie wewnętrzne polecenia nazywamy wewnętrzne polecenia Shella,44:36czyli niektóre polecenia nie powodują utworzenia nowego procesu,44:42tylko Shell je po prostu realizuje sam w ramach swojego kodu.44:46Istotne jest to, że Shell przed wykonaniem polecenia44:51musi wiedzieć, kto wykonuje polecenie i gdzie to polecenie,44:56czyli ścieżkę, mieć informację o ścieżce,44:59gdzie to polecenie jest umieszczone w systemie.45:03Typowe zachowanie to jest takie, że Shell wykonuje,45:10żeby uruchomić potomka, to wykorzystuje funkcję systemową fork,45:16wtedy typowy to jest zasypia,45:19wykonuje potem funkcję systemową wait.45:22Dla tego nowego procesu, który na początku jest w pamięci operacyjnej,45:28to jest kopia tego procesu macierzystego,45:31ale gdy wywołana zostanie funkcja systemowa exec45:34z odpowiednim argumentem,45:36to którym to argumentem jest odwołanie45:39właśnie do pewnego programu wykonywalnego,45:45no to wtedy kod tego programu znajdzie się45:48w przestrzeni adresowej tego procesu45:51i on będzie wykonywany, ten program.45:54Gdy zakończy, mówiłem o zakończeniu,45:57wtedy Shell jest budzony45:59i przykładowo wyświetla znak zachęty46:02do wydania kolejnego polecenia.46:09Tak jak wspomniałem, Shell jest to program,46:14który no można powiedzieć pośredniczy46:16między jądrem systemu operacyjnego,46:19a systemem plików46:21i w ogóle całym systemem operacyjnym można powiedzieć46:26i jego podstawową rolą jest interpretowanie poleceń.46:31Czyli użytkownik wprowadza polecenie,46:34Shell czyta to polecenie,46:37przetwarza je odpowiednio,46:40jeśli trzeba to tworzy proces46:42i następnie ten proces,46:44jako proces potomny jest wykonywany,46:47obsługiwany przez system operacyjny.46:50Tych rodzajów interpretatorów poleceń,46:54czyli tego Shella jest wiele,46:56no bo wiele zespołów programistów je tworzyło.47:00Przykładowo taki pierwszy,47:02który był tworzony to był Shell Berna.47:05Tutaj wspominałem,47:07o takim Shellem Corna,47:09który na przykład przy zamachach47:11jest używany.47:12Jest również Shell C pochodzący z Kalifornii,47:14czyli z Berklej,47:16ale jest jeszcze Bern Again Shell47:19i inne programy tych interpretatorów poleceń.47:23No ten był bardzo,47:25to był pierwszy,47:26ten Berna był bardzo ograniczony,47:28te kolejne zawierały szereg dodatkowych funkcji,47:31które no w tej chwili są stosowane,47:33wygodne dla użytkowników.47:35No i tutaj popatrzmy na takie przykłady.47:37Muszę kilka razy to przewrócić.47:44Takie podstawowe funkcje Shella,47:46które w tej chwili są dla użytkownika dostępne.47:50To podstawowe to jest właśnie47:53interpretowanie i przekazanie utworzonego procesu47:58do obsługi przez system operacyjny.48:02No ale także ma pewne polecenia,48:04są wbudowane wewnątrz,48:05czyli są obsługiwane,48:07bezpośrednio przez sam Shell,48:09ale także te współczesne Shelle,48:14aplikacje Shella,48:15no także zapewniają pewien język do pisania skryptów.48:19Pozwala także ustawić środowisko pracy użytkownika.48:26Przywoływanie i edycja uprzednio wydanych poleceń,48:29czyli to taki mechanizm historii,48:31że można sobie przywołać poprzednie,48:34wcześniej wydane polecenie i je,48:36je edytować,48:39żeby niekoniecznie napisać całe od początku.48:43Przeadresowanie czy też przekierowanie wejścia i wyjść poleceń48:48polega na tym,48:49że zamiast informacji do polecenia z terminala,48:53czyli z klawiatury,48:55można ją przesłać z pliku pewnego49:00i podobnie wyjścia polecenia,49:03takie wyjścia są,49:04jest dwa rodzaje tych wyjść,49:06wyjście takie standardowe i wyjście diagnostyczne.49:09Standardowe to na nim pojawiają się informacje generowane przez sam polecenie,49:16czyli ten program,49:17który jako proces jest uruchomiony tym poleceniem,49:21a diagnostyczne to są informacje o ewentualnych błędach,49:25tam się pojawiają.49:27No i tutaj w przypadku tego przekierowania,49:30no to mamy możliwość do dwóch plików wysłania,49:33do jednego te informacje standardowe,49:36a do drugiego te informacje diagnostyczne.49:40Generowanie na spiku przy pomocy metaznaku,49:43nie wiem, przypuszczam, że już Państwo na ćwiczeniach to ćwiczyli,49:46umożliwienie łączenia poleceń w potok,49:49czyli możliwość taką,49:51że wynik jednego polecenia jest bezpośrednio kierowany na wejście drugiego polecenia.49:58Do tego celu wykorzystywany jest odpowiednio tworzony bufor50:02w ramach tworzenia potoku między jednym,50:05a drugim procesem.50:07No i wreszcie jest taka dodatkowa funkcja,50:11możliwość przetwarzania poleceń w tak zwanym drugim planie.50:16O tym też za chwilkę kilka słów będzie.50:22Popatrzmy teraz na to,50:24w jaki sposób Shell interpretuje polecenia.50:33To jest tekst polecenia,50:35pisany przez użytkownika.50:37On wczytuje sobie do swojego bufora50:40taki ciąg znaków, bo to jest ciąg znaków dla niego.50:44Następnie na podstawie spacji50:47dzieli ten tekst na słowa,50:50czyli w tym mamy takie słowa LS, L,50:54plik ze znakiem zapytania50:56i test ze znakiem zapytania,50:58ale w cudzysłowie.51:00I teraz zaczyna51:02interpretować poszczególne słowa.51:05Ta interpretacja jest od końca.51:08Tutaj w tym przykładzie widzimy,51:10zauważył cudzysłów kończący51:13i wtedy, jeśli jest słowo objęte cudzysłowem51:19i w nim się pojawi znak specjalny,51:22którym jest znak zapytania,51:27on jest traktowany w ogólnym wypadku,51:29on może być metaznakiem,51:31ale w tym przypadku, jeśli jest objęty cudzysłowem,51:34to przypisane jest niemu51:39po prostu znaczenie znaku ASKI,51:42niespecjalne.51:46Natomiast w przypadku, gdy nie ma cudzysłowa,51:49nie ma tego, tu widzimy te cudzysłowa,51:52to on ma znaczenie specjalne,51:55czyli powoduje rozwinięcie tej nazwy.52:00W tym wypadku też sądzę,52:04że Państwo już to ćwiczyli na ćwiczeniach,52:06że w miejsce znaku zapytania52:09wstawiany jest jeden znak, ale dowolny.52:12Próbuję po kolei, to kolejne znaki,52:15jeden, dwa, trzy i tak dalej.52:17Widzimy, że w tym poleceniu LS-OPCIO-L52:20będzie chciał wyświetlić informacje52:22o tych konkretnych plikach,52:25które mają takie i takie nazwy.52:28Tu widzimy, jak rozwinął ten znak specjalny,52:33znak zapytania ten,52:35a dla niego, a tutaj ta nazwa ma,52:38jest w postaci te ze znakiem zapytania.52:42To był w cudzysłowie,52:44więc nie traktuję tego jako znak specjalny.52:47Ten pierwszy traktuję jako metaznak,52:51który należy rozwinąć.52:54I ta postać polecenia zostaje,52:58uznana jako ta do wykonania,53:02czyli uruchamia odpowiedni proces53:05z takimi argumentami, z tą opcją,53:07z tymi argumentami, czyli będzie informacja53:10o tych plikach wyświetlona.53:12Po zakończeniu wykonania tego polecenia53:15zgłasza gotowość przyjęcia nowego polecenia.53:23Teraz popatrzmy,53:25tutaj muszę znowu przesunąć parę razy.53:28Wspominałem, że są polecenia tak zwane zewnętrzne,53:35czyli takie, które wymagają utworzenia procesu53:38i wewnętrzne, obsługiwane przez Shell.53:41No i tu mamy przykłady.53:43Takie wewnętrzne, na przykład Shell,53:45no to polecenie CD, change directed,53:47czyli exit, ale tu w odniesieniu do zakończenia SOSI,53:51a tutaj przykłady poleceń zewnętrznych,53:54czyli takich, które wymagają utworzenia procesu,53:57to jest polecenie LS, MO czy inne.54:00No i oczywiście, jeśli mówimy o tych wbudowanych,54:04to ich zestaw zależy od Shell'a,54:07no i w różnych wersjach systemów aplikacji tego Shell'a54:15mogą być różne, ten zestaw może być.54:18No te standardowe zwykle są takie same.54:21Istotne jest to, że one są obsługiwane w ten sposób,54:25że jest odpowiednia, ciąg instrukcji w ramach samego Shell'a,54:31które są realizowane po podejrzeniu się takiego polecenia54:37i to przez sam Shell jest realizowane.54:41Nie tworzony jest nowy proces.54:43Natomiast, jeśli mówimy o poleceniach zewnętrznych,54:46no to wtedy właśnie tworzony jest oddzielny proces,54:49no to musi być informacja, jak je poprzez funkcję,54:55jaki program ma być realizowany przez taki nowy proces54:59i musi być podana informacja,55:03gdzie ten program, który ma być wykonywany jako proces,55:08jest umieszczony, czyli ta informacja w katalogu.55:11No i istotne jest to, że użytkownik musi mieć prawo55:15do wykonywania określonego programu jako procesu.55:20Przez skrypty rozumiemy ciąg poleceń,55:24które mogą zawierać zarówno polecenia wewnętrzne Shell'a,55:27jak i te polecenia zewnętrzne.55:29I tu również użytkownik musi pamiętać o tym,55:32żeby taki skrypt miał prawo do czytania i do wykonywania.55:41A teraz popatrzymy na,55:45no musimy troszkę w prawo przerzucić,55:48na dwa sposoby wykonywania poleceń.55:50Jeden sposób to jest tak zwane wykonywanie synchroniczne,55:54albo też mówimy przetwarzanie w pierwszym planie,55:57w angielsku foreground,56:00w którym działa w sposób standardowo,56:04czyli tak, Shell tworzy nowy proces,56:08ale wcześniej kiedy tworzy?56:11No zgłosił gotowość przyjęcia polecenia pewnego,56:15użytkownik wpisał polecenie, które wymaga utworzenia procesu,56:19wtedy tworzy nowy proces,56:23i usypia.56:25Czeka na jego zakończenie.56:27A ten nowy proces, jako nowy proces, no się wykonuje.56:34Gdy zakończy się, no wspomniałem co się wtedy dzieje,56:41system operacyjny wtedy budzi Shell,56:45wtedy dopiero proces znika z tablicy procesów,56:49a Shell wyświetla znak zachęty,56:54został obudzony i zaczyna działać.56:59Pozwala wprowadzić kolejne polecenia.57:02Ale jest jeszcze drugi mechanizm wykonywania procesów,57:07w tak zwanym drugim planie, po angielsku background,57:10albo też asynchronicznym.57:13I tutaj też Shell zgłasza gotowość przyjęcia polecenia,57:18jeśli użytkownik wpisze polecenie, no to się znajduje,57:23interpretuje, jak wspomniałem wcześniej,57:25ale tworzy nowy proces, ale nie czeka na jego zakończenie.57:30Czyli od razu wyświetla znak zachęty,57:34czyli możliwość przyjęcia kolejnego polecenia.57:37Czyli jeszcze poprzedni proces utworzony w ten sposób57:43przez użytkownika, no jego poleceniem, już działa.57:47Jeszcze się nie skończył, a on może wywołać kolejny proces.57:51Czyli w tym przypadku, co jest istotne,57:54że ten Logi Shell może mieć wiele procesów potomnych,57:59które no właśnie tworzą się, no kończą się,58:04o tym też Shell będzie informowany.58:07No istotne jest wtedy, nie wiem czy Państwo już to ćwiczyli na ćwiczeniach, czy nie,58:12istotne jest wtedy właśnie wykorzystywanie tego polecenia procesu,58:17tego statusu, żeby wiedzieć jaki jest stan poszczególnych procesów w systemie.58:23No istotne jest to przekierowanie informacji wejściowej i wyjściowej,58:28żeby informacja do polecenia trafiała z pewnego pliku,58:32a informacje wyjściowe, żeby nie mieszały się na tym terminalu,58:37tylko żeby do odpowiednich plików były wpisywane.58:41Ale tu szczegóły też są na ćwiczeniach.58:47Popatrzmy teraz na przykłady działania interpretatora poleceń.58:55Tutaj kilka przykładów Państwu przedstawiłem.59:00Czyli rozpatrzmy takie typowe działanie,59:04że interpretator wczytuje wiersz polecenia z pewnego terminala,59:10z pliku standardowego i interpretuje go.59:14Mamy takie polecenia proste jak59:16hu czy les, no to wtedy, tak jak wspomniałem,59:19wykonuje funkcję systemową fork, następnie funkcja on usypia,59:25a dla tego nowego procesu wykonywana jest funkcja exec59:29z odpowiednim argumentem wynikającym z treści polecenia59:35i jest wykonywany program, który tutaj ma być zrealizowany.59:46A sam szel w tym czasie jest uśpiony, czeka na zakończenie,59:52a jak zostanie ten proces potomny zakończony,59:56no to wyświetla zachętę, jak wspomniałem wcześniej,59:59i pozwala wpisać kolejne polecenie, no i wtedy znowu je obsługiwa.1:00:03Ale są też polecenia wbudowane,1:00:09także jak chceć dyrektor, czy fork, czy while1:00:12do pisania skryptu, te dwa ostatnie,1:00:14to wtedy interpretator nie tworzy dla ich wykonania nowych procesów,1:00:20ale sam je obsługuje w ramach swojego kodu.1:00:24Teraz popatrzmy na przykład polecenia ze zmianą standardowego pliku wyjściowego.1:00:30Jako przykład podaję polecenie ls z opcją l, strzałka, lista.1:00:35No widzimy, że to będzie, to się odnosi do katalogu bieżącego1:00:39w danym momencie, czyli celem tego polecenia1:00:44jest wyświetlenie aktualnie istniejących w danym katalogu bieżącym1:00:50plików z ich informacjami, o nich.1:00:53Jak to się dzieje? Najpierw proces potomny tworzy plik, lista,1:00:58no może próba się nie powiedzieć,1:01:01no to wtedy nie wykona zadania, no jak nie będzie wolnego i wezwa,1:01:04to nie utworzy nowego pliku.1:01:07Jak się nie uda, to zakończy działanie,1:01:10a jeśli zostanie utworzony, to wtedy1:01:13zamyka standardowy plik wyjściowy, którym jest terminal, czyli ekran,1:01:19a powiela deskryptor tego właśnie pliku o nazwie lista1:01:24i wpisuje, ja następnie wykonuję to polecenie ls z opcją l1:01:28i wpisuję te informacje do tego pliku listy.1:01:32Kończy działanie.1:01:36Tutaj popatrzmy, jak to wygląda w przypadku wykonywania polecenia w tle.1:01:43Żeby wykonać pewne polecenie w tle,1:01:46używamy takiego znaku ampersand,1:01:50to jest znacznie znak przetwarzania w tle1:01:53i jak wspomniałem, to interpretacja jest od końca,1:01:57czyli gdy napotka ten znak ampersand,1:02:01to wie, że ma być to przetwarzane w tle1:02:04i ustawia odpowiednią zmienną lokalną,1:02:08oznaczającą właśnie, że jest to wykonywane, to polecenie w tle.1:02:13Istotne jest to, że gdy pod koniec swojej pętli sprawdza wartość tej zmiennej1:02:20i jeśli jest ustawiona,1:02:22to nie wykonuje tej funkcji wait, czyli uśpienia,1:02:27ale wraca do stanu gotowego1:02:30i pozwala wczytać następne polecenie,1:02:34czyli wyświetla znak zachęty,1:02:36czyli nie czeka na zakończenie polecenia tego procesu.1:02:43Potomnego, bo proces wykonujący to polecenie greb1:02:48będzie potomnym względem Shell'a,1:02:52ale nie czeka na jego zakończenie.1:02:57A teraz popatrzmy na jeszcze jeden przykład.1:03:00Polecenie z potokiem.1:03:02Tutaj przygotowałem taki dodatkowy1:03:07plik z interpretacją.1:03:13Mamy tak, polecenie LS opcją L,1:03:19znak potoku, czyli pionowa kreska,1:03:22a kolejne polecenie word count,1:03:24które zlicza liczbę znaków, liczbę słów, liczbę wierszy.1:03:30Interpretacja jest od końca.1:03:32Czyli najpierw tworzy, wykorzystując funkcję fork,1:03:36a potem exec, proces potomny word count.1:03:40Ten proces potomny tworzy potok,1:03:43zgodnie z kolejnością interpretacji,1:03:46i wywołuje funkcję fork, tworząc swojego potomka.1:03:50Czyli ten proces LS z opcją L,1:03:56realizujący to polecenie,1:03:59będzie potomkiem, ale nie Shell'a,1:04:02ale tego word countu.1:04:04I wtedy dopiero wnuk interpretatora1:04:10wykonuje pierwszą,1:04:13część polecenia LS,1:04:16ale teraz musi wpisać to do potoku.1:04:20I to popatrzmy, jak to się dzieje jeszcze raz,1:04:22na tym, co zostało w sfermaciku.1:04:25Czyli mamy Shell,1:04:27który utworzył proces word count.1:04:30Ten word count utworzył potok,1:04:34a następnie utworzył swojego potomka.1:04:38Też zatknął, jak ten potomek zacznie działać.1:04:42Ten potomek ma wpisać teraz do potoku1:04:48tą informację wygenerowaną.1:04:52Wtedy usypia, budzi word count.1:04:55Word count czyta z tego potoku informację.1:04:59No i kończy działanie.1:05:03Ta informacja zostanie sprawdzona,1:05:06co do liczby znaków, słów, wierszy.1:05:09Czyli tu widzimy,1:05:11istotne jest też automatyczne1:05:14to przekierowanie informacji, prawda?1:05:16Czyli ten standardowy plik wejściowy1:05:19to jest terminal, czyli klawiatura.1:05:21A on ma sczytać to z potoku do czytania.1:05:26Po przeczytaniu zamyka deskę do potoku,1:05:29wykonuje polecenie word count.1:05:32No i wtedy budzony jest Shell1:05:36i pozwala kolejne polecenie wpisać.1:05:41Jest to dla Państwa...1:05:44No ja szybko muszę niestety mówić,1:05:46czy jest to...1:05:48czy złapali Państwo ten sens tego tworzenia.1:05:52Istotne jest to właśnie1:05:54interpretowanie od końca1:05:56i tworzenie w związku z tym1:05:58w odpowiedniej kolejności tych procesów,1:06:00których...1:06:02Czyli jeszcze raz.1:06:04Shell był tutaj.1:06:06On utworzył swojego potomka,1:06:08którym był word count i w tym czasie usnął.1:06:10Word count utworzył jako swojego potomka LSL1:06:13proces wykonujący to polecenie.1:06:15Też usnął.1:06:18Gdy LS wpisze to do potoku,1:06:21to co wygenerował.1:06:24W momencie, gdy on zakończy działanie,1:06:27to system budzi word count,1:06:29który odczyta potok.1:06:33Tu świetnie informacje o tym...1:06:36inform... dotyczące tego...1:06:38tej liczby...1:06:39tej liczby znaków, słów, wierszy.1:06:42Skończy działanie.1:06:45Exit.1:06:47No to wtedy zostanie przez systemu...1:06:49Informacja o tym trafi do Shella,1:06:51który zostanie obudzony.1:06:53I tu widzimy, jak te kolejne procesy potomne są tworzone,1:06:57a potem jak kończą wykonanie.1:07:01Przepraszam, mam pytanie.1:07:03Proszę bardzo.1:07:05A jeżeli chodzi o to budzenie Shella,1:07:07to czy to jest właśnie spowodowane...1:07:09W sensie...1:07:11Czy to jest tak, że właśnie jak wykonujemy jakąś komendę,1:07:13która no... możemy zauważyć,1:07:15że trwa na tyle długo,1:07:16to dlatego nie mamy jakby dostępu do...1:07:18Tak, dokładnie.1:07:19Terminala?1:07:20I potem możemy na przykład przerwać te procesy1:07:22w tle na przykład CTRL-C,1:07:23i wtedy mamy jakby...1:07:25No CTRL-C, to pamiętają Państwo,1:07:27mówiłem o tym wcześniej.1:07:29Wysyła... to jest no...1:07:31podpełnik polecenia KIL, prawda?1:07:33No... tak.1:07:35No tak, na tych procesach potomnych Shella, tak?1:07:37Tak, ok.1:07:38Tak, dokładnie.1:07:39Czyli no...1:07:40No to...1:07:41To...1:07:42Tylko o tym mówiłem właśnie.1:07:46Informacja o zakończeniu procesu, prawda?1:07:49Czyli proces, gdy się kończy,1:07:51to system operacyjny musi1:07:54przekazać informację o tym zakończeniu.1:07:57On już jest zakończony,1:07:59ale jest ciągle zombie.1:08:00Dopiero jak system operacyjny przekaże to do Shella,1:08:04no to Shell wtedy będzie się budził1:08:06i dalej będzie działał.1:08:08Także jak są polecenia,1:08:09które wymagają bardzo długiego czasu wykonywania,1:08:12no to trzeba czekać na możliwość wykonania polecenia,1:08:16jeśli to będzie w tym realizowane,1:08:19tak jak to polecenie w charakterze background, prawda?1:08:25Czyli bezpośrednio.1:08:27Natomiast jeśli będzie do tego celu właśnie1:08:30przy wykonywaniu poleceń,1:08:31które wymagają bardzo długiego czasu,1:08:34warto użyć tego znaku,1:08:37o którym tutaj wspomniałem,1:08:44ampersand,1:08:45bo wtedy to polecenie będzie wykonywane w tle1:08:48i pozwala,1:08:49ono sobie będzie jako oddzielny proces wykonywane,1:08:52a Państwo mogą kolejne polecenie po prostu wydać1:08:55do wykonywania na bieżąco.1:08:58Czyli w sumie, jeżeli na przykład będę chciał wykonać1:09:01polecenie update,1:09:04czyli dodam ten znak na końcu,1:09:06to jakby w sensie nie będę...1:09:07To zostanie utworzony proces,1:09:09ale shell od razu będzie działał.1:09:11Tak, ale będzie też widział, co jest tam instalowane, prawda?1:09:15A to wtedy, jak wykona Pan polecenie proces status,1:09:19to będzie Pan mógł oglądać, co tam się z nim dzieje.1:09:22A, okej.1:09:23Ale wtedy trzeba pamiętać,1:09:25że to wynik tego polecenia,1:09:28które będzie w formie background,1:09:31czyli w tle wykonywane,1:09:32musi Pan przekierować wyniki,1:09:35czyli dokonać tego przekierowania informacji, prawda?1:09:39Do jakiegoś pliku,1:09:40no bo tak Panu się wszystko tam będzie mieszało na ekranie.1:09:43Także na to zwracam uwagę.1:09:45Ale rozumiem, że Państwo jeszcze tego nie ćwiczyli na ćwiczeniach, tak?1:09:49Czy już było to?1:09:51No czy przekierowanie pliku było?1:09:54Nie pamiętam.1:09:56A ta praca w tle?1:09:58To też było, też było.1:10:00Aha.1:10:01No to w zasadzie ja tylko takie uzupełniające,1:10:04jak to wygląda z punktu widzenia systemu operacyjnego,1:10:06to przedstawię.1:10:08Dobrze.1:10:09Możemy iść dalej, tak?1:10:11No to parę słów o koncepcji wątków.1:10:13Co rozumiemy przez wątek?1:10:16Otóż oprogramowanie można pisać w różny sposób.1:10:19Można pisać to,1:10:21żeby wykonać pewne zadanie informatyczne,1:10:24jako pewien ciąg,1:10:26jako program jeden,1:10:28który zawiera pewien,1:10:30pewną sekwencję,1:10:31pewien ciąg rozkazu.1:10:33Ale można to oprogramowanie napisać jako1:10:35nie jedną sekwencję rozkazów,1:10:37ale kilka sekwencji rozkazów1:10:41współpracujących ze sobą,1:10:45które wspólnie wykonują,1:10:48no odpowiednio się przekazując między sobą informacje,1:10:51wykonują dane zadanie informacyjne.1:10:53I wtedy mówimy o,1:10:55używamy takiej nazwy,1:10:57proces wielowątkowy, który1:10:59obsługuje realizację tych kilku sekwencji.1:11:03Czyli każda sekwencja oddzielna rozkazów1:11:05jest traktowana jako oddzielny wątek.1:11:08Ale istotne jest to,1:11:10że zadanie informatyczne, które one wspólnie realizują,1:11:13no jest jedno.1:11:17Bo istotne jest to, że te wątki mogą być1:11:19współbieżnie wykonywane,1:11:21czyli na zmianę mogą dostawać kwanty czasu procesu.1:11:24I czasami używa się takiego1:11:27sformułowania proces lekki,1:11:28czyli możemy mówić tak,1:11:32o procesach jednowątkowych1:11:35i procesach wielowątkowych.1:11:37Istotne jest to, że wątek,1:11:41wątki może inaczej,1:11:43wątki w ramach jednego procesu wielowątkowego,1:11:46czyli takiego oprogramowania,1:11:48które realizuje jedne wspólne zadanie informatyczne,1:11:54mają wspólną przestrzeń,1:11:56wspólną przestrzeń adresową,1:11:59to jest bardzo inne.1:12:02Czyli w ramach tej wspólnej przestrzeni adresowej,1:12:06on właśnie będzie dzielił te informacje,1:12:10które tutaj są zasoby systemu,1:12:12czyli szereg informacji dla tych wszystkich wątków1:12:15będzie ten sam, wspólny,1:12:17bo będzie wspólna przestrzeń adresowa.1:12:19Natomiast każdy jest z oddzielną sekwencją rozkazów,1:12:23czyli każdy musi mieć swój rozkaz,1:12:25list licznych rozkazów,1:12:27stan rejestrów,1:12:29no bo tego, który trzeba zapisywać w jego bloku kontrolnym,1:12:33każdy będzie miał swój blok kontrolny,1:12:35gdzie będzie ten, Państwo pamiętają,1:12:37że przy przełączaniu procesora1:12:39trzeba zapisać stany rejestrów1:12:41do tego bloku kontrolnego.1:12:43Czyli każdy wątek ma pewne informacje niezależne,1:12:48ale przestrzeń adresową całą ma wspólną1:12:51z innymi realizującymi to samo zadanie.1:12:55Teraz, a tradycyjny, czyli taki jednowątkowy proces,1:13:01mówimy wtedy czasami proces ciężki,1:13:04to każdy posiada oddzielny obszar przestrzeni adresowej,1:13:08odpowiada takiemu zadaniu właśnie z jednej wątki.1:13:11Natomiast w przypadku oprogramowania wielowątkowego,1:13:14czyli takiego procesu wielowątkowego,1:13:17mamy dzielenie wspólnych zasobów1:13:20w ramach tego zadania informatycznego.1:13:23I problem polega na tym,1:13:25że przełączanie wątków w ramach tego jednego zadania1:13:30jest znacznie, tu napisałem, znacznie mniej kosztowne.1:13:33W jakim sensie?1:13:34Mniej informacji trzeba zapisywać w bloku kontrolnym1:13:37i je potem odtwarzać.1:13:39Jak mniej zapisujemy i odtwarzamy,1:13:42to mniej czasu to zajmuje,1:13:44czyli to, co nazywamy czasem przełączania kontekstu,1:13:48jest po prostu krótsze.1:13:50Czyli system działa efektywnie.1:13:53Ale ja pokażę jeszcze Państwu,1:13:55że w tym roku w ramach systemów rozproszonych takiego kursu,1:14:00że także więcej korzyści z pracy wielowątkowej można uzyskać.1:14:07Tam będą dalsze informacje dotyczące pracy wielowątkowej.1:14:12Czyli one działają, wątki działają współbieżnie na zmianę,1:14:17no właśnie, w podobnych mogą być stanach jak procesy,1:14:21na zmianę dostają kwanty czasu procesorów.1:14:23No ale jak mamy jeden procesor,1:14:25no to będzie tylko w danym momencie,1:14:27tylko jeden wątek mógłby być kontynuowany.1:14:29Ale nawet wtedy praca wielowątkowa1:14:33zapewnić może znacznie większą efektywność działania systemu.1:14:38To jest istotne.1:14:40Na czym polega to, że one nie są niezależne?1:14:43One są w jakimś sensie zależne od siebie1:14:48w ramach określonego procesu wielowątkowego,1:14:51bo mają ten sam obszar pamięci operacyjnej1:14:55i mogą z tego obszaru korzystać.1:14:58No i co jest istotne, jeśli to pochodzą od tego samego użytkownika,1:15:04to nie wymagają takiej samej ochrony,1:15:06jak procesy pochodzące od różnych użytkowników.1:15:10Godzina 13, to ja powinienem chyba teraz,1:15:17a nie, 13.15 skończymy.1:15:21Przejdziemy teraz do zagadnień planowania1:15:28czy też szeregowania procesu.1:15:31Można by powiedzieć, że jest to najpoważniejsze zagadnienie1:15:35dotyczące zarządzania procesami.1:15:40Jak wspomniałem w tym wykładzie wstępnym,1:15:44planowanie czy też szeregowanie polega na tym,1:15:48że system operacyjny będzie decydujący,1:15:51decydował o tym, jeśli mamy wiele procesów1:15:54w systemie współbieżnie działających,1:15:57w jakiej kolejności będą dostawały czas procesu.1:16:01I dotyczy to w szczególności procesów,1:16:04które są w stanie gotowym,1:16:06bo tylko takie procesy mogą dostać czas procesora,1:16:10bo są gotowe do wykonywania.1:16:14Czyli tak jak wspomniałem i tutaj napisane,1:16:18planowanie polega na określeniu,1:16:20w jakiej kolejności procesy uzyskają dostęp1:16:23do zasobów komputera, przede wszystkim procesora,1:16:26ale także innych zasobów, pamięci operacyjnej.1:16:30Może być na przykład 5 procesów, które chcą1:16:33wykonania operacji dyskowej.1:16:36To też będą musiały być ustawione w pewnej kolejce,1:16:40czyli będzie musiało być ten zagadnień1:16:43szeregowania rozwiązany.1:16:46Można to zrobić w różny sposób.1:16:49No więc tu chodzi ogólnie,1:16:51można powiedzieć, że zależy nam,1:16:53żeby jak najlepiej wykorzystać procesor,1:16:55żeby najsprawniej system pracował.1:17:02W ogólnym przypadku mamy wiele takich kolejek.1:17:07Kolejek całych zadań informatycznych,1:17:10są kolejki do procesów gotowych do procesora,1:17:14jak wspomniałem, są kolejki do urządzeń wejścia, wyjścia.1:17:19No przykładowo, gdy proces zarządza kolejki1:17:22operacji wejścia, wyjścia, no to będzie1:17:25umieszczony w kolejce do konkretnego urządzenia1:17:30tego, aby wykonać tą operację wejścia lub wyjścia.1:17:33Czyli odczyt lub zapis na dysk.1:17:36Może utworzyć nowy proces i czekać na jego ukończenie.1:17:40Może wykorzystać cały kwant czasu,1:17:43lub też może wystąpić pewne przerwanie1:17:46i zostać przeniesiony.1:17:49Do kolejki procesów gotowych.1:17:52Tutaj mamy takie przykłady różnych operacji,1:18:00z którymi związany może być ten problem przeregowania.1:18:05Mamy procesor, mamy pewną liczbę procesów gotowych,1:18:15do którego1:18:19do tego zestawu procesów gotowych,1:18:22czyli do tych nowych rezultatów musi być pewna kolejka utworzona1:18:26przez system operacyjny.1:18:29Zarówno trafiają nowe procesy,1:18:31jak i te procesy, które wykonywały się już wcześniej1:18:35i potem po jakimś czasie z określonych powodów1:18:38wracają do tej kolejki.1:18:40O tym należy pamiętać, że w tej kolejce1:18:43mamy zarówno nowe, jak i te, które już wcześniej działały,1:18:46tylko z jakichś powodów zostały z procesora wyrzucone,1:18:48i wrócimy do kolejki procesów gotowych.1:18:52No i oczywiście, jeśli proces jest zakończony,1:18:55no to już kończy działanie.1:18:58No a teraz popatrzmy, jakie to mogą być kolejki.1:19:02Może być sytuacja taka, że proces zażądał operacji wejścia-wyjścia,1:19:08no to zostanie ustawiony w kolejce do tego urządzenia,1:19:12uśpiony jest, bo to obsługuje system operacyjny,1:19:16gdy ta operacja wejścia-wyjścia zostanie wykonana,1:19:20no to proces będzie budzony i wróci do kolejki procesów gotowych.1:19:24Proces zużył kwant czasu, no to został wyrzucony z procesora,1:19:29no i powrócił do kolejki procesów gotowych.1:19:33Proces czeka na pewne przerwanie,1:19:35na przykład na wpisanie pewnych informacji,1:19:38na przykład z klawiatury, czy z jakiegoś innego powodu.1:19:44No to aż to zostanie.1:19:46Jeżeli to zostanie zrealizowane,1:19:49no to wróci do kolejki procesów gotowych.1:19:53Proces powołuje potomka, prawda?1:19:56No to zostaje uśpiony, tak jak szer w standardowym działaniu.1:20:00Potomek się wykonuje, potomek się skończy,1:20:05to ten szer, ten proces dostaje,1:20:08no ale to dotyczy wszystkich procesów, prawda,1:20:10które utworzyły potomka.1:20:13To wtedy ten proces macierzysty jest budzony1:20:17i wraca do kolejki procesów gotowych.1:20:20No i to samo dotyczy zagadnień związanych z pamięcią operacyjną,1:20:25no bo nie wszystkie przykładowo procesy1:20:27mieszczą się w pamięci operacyjnej,1:20:29więc tu też może być sytuacja,1:20:31że proces zostanie przerzucony z pamięci operacyjnej1:20:35na obszar wymiany słop,1:20:37a dopiero później wróci do wykonywania do kolejki procesów gotowych.1:20:43Czyli mamy, widzą Państwo, wiele powodów,1:20:46dla których te procesy, można powiedzieć, tak krążą1:20:51między procesorem i kolejką procesów gotowych.1:20:54Przy czym ten diagram dotyczy, no,1:20:58bieżących systemów operacyjnych,1:21:00w których mamy stosowany podział1:21:04czasu procesora na pewne kwanty.1:21:07Można różnego rodzaju, no,1:21:13mamy kryteria planowania,1:21:15o, to trzeba kilka razy znowu przewrócić.1:21:23Teraz tak.1:21:25To realizuje system operacyjny.1:21:29To zagadnienie szeregowania można w różny sposób rozwiązać,1:21:33jak wspomniałem.1:21:35No i mamy teraz, chcemy, żeby, no,1:21:38to działanie systemu było jak najefektywniejsze.1:21:42No to, co znaczy najefektywniejsze?1:21:45No, można rozpatrywać różne kryteria planowne.1:21:48No, jedno takie kryterium to jest1:21:51procent wykorzystania czasu procesu.1:21:54Czyli czas biegnie, procesor, no nie,1:21:58no, jeśli proces w danym momencie się przykładowo1:22:02został uśpiony z powodu czekania na operację dyskową,1:22:06no to w tym czasie, a jest tylko jeden, no to1:22:10system, to procesor w tym czasie, no nie, nie działa, prawda?1:22:14W tym sensie, że nie obsługuje konkretnego procesu.1:22:17Chcemy, żeby jak najwięcej czasu procesora było wykorzystane1:22:22na rzeczywiście wykonywanie procesu, prawda?1:22:26No, także w czasie przełączania kontekstu również1:22:29procesor nie realizuje żadnego procesu.1:22:33Czyli proces czasu, no to czas, w którym proces obsługuje pewne procesy,1:22:40to procesy do całego czasu, na przykład do całego dnia.1:22:44Przepustować, no można sobie wyobrazić kryterium tego typu,1:22:48że w ciągu na przykład doby policzymy ile1:22:54procesów zostało wykonanych, zakończonych.1:22:59No, można to wziąć taką średnią, tam powiedzmy z miesiąca1:23:03i powiedzieć, że to będzie średnio w ciągu jednej doby tyle i tyle1:23:07procesów ukończonych, to będzie przepustować.1:23:10Oczywiście tu musi być mowa o pewnych wartościach średnich,1:23:14bo procesy są dłuższe, krótsze, więc ich jest więcej rozwinięć.1:23:18Czas cyklu przetwarzania.1:23:21To jest dla pojedynczego procesu, to jest istotne,1:23:26Państwo sobie to zapiszą, to jest czas od rozpoczęcia procesu,1:23:31proszę zapisać, to jest czas cyklu przetwarzania.1:23:35Czas od rozpoczęcia procesu do jego zakończenia.1:23:38W tym czasie,1:23:40są przedziały czasu, gdy proces się wykonuje1:23:44i są przedziały czasu, gdy proces czeka, na przykład w stanie gotowym1:23:48czy w innym stanie, na przykład w stanie uśpienia.1:23:52Czas od zgłoszenia procesu, jego utworzenia do jego zakończenia.1:24:00Ale można zrobić na przykład kolejny czas oczekiwania,1:24:04czyli taki średni czas oczekiwania w kolejkach.1:24:08No a tutaj w ogólnym wypadku też trzeba mówić o pewnej średniej wartości1:24:12tego czasu cyklu przetwarzania dla różnych procesów.1:24:16No oczywiście chcemy, żeby wykorzystania procesora było jak największe,1:24:23żeby przepustowość była jak największa,1:24:26a chcielibyśmy, żeby ten średni czas cyklu przetwarzania był z kolei jak najmniejszy1:24:31i też żeby średni czas oczekiwania w kolejkach był jak najmniejszy.1:24:36A tu jeszcze jedno kryterium wpisałem.1:24:39Czas odpowiedzi.1:24:41To dotyczy systemów takich interakcyjnych,1:24:45czyli aplikacji takich jak na przykład edytor.1:24:49No bo włączymy sobie ten edytor i działamy.1:24:53Czasami coś robimy, czasami sobie zrobimy herbatę.1:24:57No to czas, policzenie czasu ile ten edytor w ogóle działa,1:25:03no to nie jest najlepszym kryterium.1:25:06Ale istotne jest wtedy to,1:25:08no bo to zależy od tego jak długo użytkownik tylko włączył, no jak długo.1:25:13Ale jak szybko, czas odpowiedzi to jest jak szybko dostanie odpowiedź1:25:19przy zgłoszeniu określonego działania do realizacji przez daną aplikację.1:25:29Dobrze, a teraz popatrzmy na pewne przykłady algorytmów szeregowania1:25:35czy też planowania.1:25:37W tej chwili nie mówię o konkretnym,1:25:40Państwo się zadają, że to nie jest mowa o konkretnym systemie operacyjnym,1:25:44czyli na przykład o systemie Nixowym czy innym,1:25:48ale o różnych możliwych, a także stosowanych historycznie,1:25:54ideach, algorytmów planowania.1:26:02To już będzie niepotrzebne nam.1:26:05Popatrzmy na taki pierwszy przykład.1:26:07To jest taki, czyli problem dotyczy ustawienia tych procesów w pewnej kolejce,1:26:13prawda, według której będą dostawały czas procesora.1:26:16Czyli mamy przykładowo pewną liczbę procesów w stanie gotowym1:26:21i system operacyjny decyduje teraz, który z nich wybrać jako pierwszy,1:26:26jako drugi, jako trzeci.1:26:28Ten pierwszy algorytm to jest first come, first served.1:26:33Czyli można powiedzieć,1:26:34taki, który jest obsługiwany przez kolejkę first in, first out,1:26:40czyli który wcześniej przykładowo znalazł się w tym stanie gotowym,1:26:44no to pierwszy nadszedł, no to pierwszy będzie obsłużony.1:26:48Czyli ta kolejka FIFO to jest forma określenia kolejności ich obsłużenia.1:26:56Natomiast on ma pewne wady i sprawia pewne kłopoty.1:27:02I tu popatrzmy, o, o widzę, że pan Rzeszatko mi coś przysłał.1:27:07Dziękuję.1:27:08Dla dziesiątki.1:27:10Popatrzmy teraz, tutaj też przygotowałem tutaj pewien przykład1:27:15ilustrujący problem.1:27:22Na czym polega ten efekt konwoju, o którym tutaj mówię?1:27:26No wyobraźmy sobie, że mamy takie trzy procesy.1:27:30Najpierw był zgłoszony,1:27:32proces A, który wymaga tysiąca sekund wykonania,1:27:35proces B jedną sekundę, proces C jedną sekundę.1:27:38I one były kolejno zgłaszane w zero, jeden w drugiej sekundzie.1:27:45Tu mamy oś czasu zgłoszenia procesu.1:27:48No jeśli zastosujemy ten algorytm,1:27:53no to powiedzmy tak, no w pewnym momencie proces wcześniej był jeszcze zajęty1:27:57przez inne wcześniejsze procesy.1:27:59No to w kolejce będą,1:28:02te trzy procesy A, B i C.1:28:04No to system operacyjny wybierze najpierw ten pierwszy,1:28:07który zajmie tysiąc sekund.1:28:10No potem kolejno zgłoszony ten drugi, potem zgłoszony ten trzeci.1:28:14Czyli przykładowo, jak byłoby to przetwarzanie jakiejś bazy danych,1:28:20a drugi użytkownik by zgłosił polecenie, część dyrektory,1:28:24no to proszę bardzo,1:28:26no to na odpowiedź na to część dyrektor będzie czytał, czekał1:28:30ten tysiąc sekund.1:28:31Bo ten wcześniejszy musi się wykonać.1:28:34To jest ten efekt konwoju.1:28:36Ten, który działa najwolniej, najdłużej,1:28:39no powoduje odpowiednie opóźnienie innych procesów.1:28:44No dlaczego efekt konwoju?1:28:46No tak, ja cały czas mówię o braku wielowodnych.1:28:52Ja cały czas mówię o procesach jednowątkowych tutaj.1:28:56Także na razie mówmy sobie.1:28:59A w przypadku wątków, no to będzie, to dotyczy kolejnego wątku.1:29:07Ja mówię, ale żeby była jasność.1:29:09Mówimy o, nie mówimy o procesach, o systemach operacyjnych1:29:14z możliwością pracy wielowątkowej.1:29:16Mówimy o algorytmie FCFS, żeby była jasność.1:29:19On jest algorytmem niewywłaszczającym.1:29:25O tym mówimy, o tej idei,1:29:28no ja ją przedstawiam zgodnie też z literaturą,1:29:34no dla porównania innych.1:29:36To był pierwszy taki historyczny, najprostszy algorytm.1:29:42No to poszukiwano wtedy algorytmów bardziej efektywnych.1:29:46No i kolejny to SJF.1:29:50Czyli znowu mamy taką kolejkę procesów gotowych.1:29:54No i wtedy w chwili spójnej,1:29:58sprawdzania tej kolejki,1:30:00system operacyjny wybiera spośród nich1:30:04taki, który najkrótszy czasu wymaga.1:30:08Czyli procesor jest przydzielany procesowi,1:30:12który będzie najmniej czasu wykonywał.1:30:16To jest algorytm teoretyczny.1:30:18Dlaczego? No za chwilkę.1:30:20On teoretycznie daje minimalny, średni czas oczekiwania.1:30:25Ale dlaczego jest teoretyczny?1:30:27Bo wymaga dokładnego oszacowania,1:30:29ile proces potrzebuje procesorom.1:30:32Czego w pierwszej chwili nie można zrobić, prawda?1:30:35Natomiast ten algorytm jest istotny1:30:37dla porównywania różnych wersji algorytmów1:30:41pod względem ich efektywności.1:30:45No można po prostu, są rozwiązania jak można szacować,1:30:49ile ten proces wymaga,1:30:52można w ten sposób to liczyć.1:30:55Teraz pojęcie,1:30:57algorytm wywłaszczający to jest taki,1:31:05który pozwala na to,1:31:08aby system operacyjny wyrzucił proces z procesora.1:31:13To jest wywłaszczający.1:31:15No wywłaszczył go z procesora.1:31:17Algorytm niewywłaszczający to jest taki,1:31:21w którym procesor,1:31:25jeśli zostanie przydzielony procesowi,1:31:28to na tyle czasu, ile on tego czasu potrzebuje.1:31:32Państwo sobie to zapiszą,1:31:34bo to jest jeden z celów tego wykładu,1:31:37żeby państwo pojęcia podstawowe1:31:40sobie w jakiś sposób ugruntowali.1:31:43Czyli możemy mówić o algorytmach wywłaszczających lub nie.1:31:46No ten FCFS ogólnie jest algorytm niewywłaszczającym.1:31:51Ten może być wywłaszczający, ale nie musimy.1:31:54Kolejny przykład to jest planowanie priorytetowe.1:32:00Czyli idea polega na tym,1:32:02że każdy proces miał pewien priorytet,1:32:06a procesy teraz będą wybierane zgodnie z tymi priorytetami.1:32:13I znowu, czyli można by powiedzieć,1:32:19że wybierany jest proces w najwyższym priorytecie wtedy,1:32:22w spośród tych procesów gotowych w danym momencie czasu1:32:27przez system operacji.1:32:29No i SJTF jest takim szczególnym przypadkiem,1:32:32gdzie im krótszy wymagany czas,1:32:37to można powiedzieć, że priorytet jest wyższy.1:32:40Widzę, że jest 13.00, a już 18.00,1:32:43to ja w tej chwili zrobię przerwę na te 15 minut,1:32:47żebyśmy wszyscy sobie troszkę odpoczęli,1:32:50po tych 15 minutach.1:32:52Ja teraz zamknę nagrywanie, a nowe nagranie zrobię