**Systemy operacyjne 1 09.12.2023 2 czesc**0:00Tak. Dobrze. Wspomniałem o tym, że jeden z panów zapytał,0:05czy jest, skoro tu jest mowa o tych szacowanym czasie procesora0:12wymaganego przez proces, czy jest możliwość uzyskania informacji0:15o ile dane polecenie będzie jeszcze trwało. No bo powiedziałem,0:20że nie znam takiej możliwości, bo jest to praktycznie niewykonalne0:27i to jest jedna rzecz, bo nie można dokładnie oszacować0:31ile czasu dany proces potrzebuje w przyszłości, a poza tym,0:37jeśli mamy wiele procesów w systemie, no to będzie zależało0:40od obciążenia systemu. A następna rzecz, że z tych właśnie względów,0:45tych algorytmów SJF, ponieważ jest trudno, praktycznie nie można0:49oszacować przyszłego czasu, praktycznie się nie stosuje.0:53Natomiast one stanowią pewien punkt odniesienia dla innych,0:57innych algorytmów, bo gdyby je można zastosować, to byłby to1:01najefektywniejszy algorytm. Istotne jest to, że ten algorytm1:06może być wywłaszczający lub nie. Ale to chyba przed przerwą mówiłem,1:11żeby państwo zapisali sobie, jaka jest różnica algorytmu1:15wywłaszczającego lub nie. Zapisali państwo, czy powtórzycie?1:21Zakładam, że zapisali. No wywłaszczający to taki,1:27że system operacyjny może w pewnym momencie czasu wyrzucić proces1:31z procesora, a niewywłaszczający, że jeśli przydzielił mu czas1:35procesora, to na tyle czasu, ile procesor potrzebuje.1:39Planowanie priorytetowe, to każdy proces dostaje pewien priorytet.1:43No gdyby algorytm SJF był możliwy do zastosowania, to im krótszy proces,1:52to miałby większy priorytet.1:55Planowanie rotacyjne.1:57To stosowane we współczesnych systemach operacyjnych,2:01w których mamy podział czasu procesora na kwanty.2:05Tu widzimy rozmiar czasu tego kwantu. Ile czasu trwa taki kwant?2:11No to rzędu 100 do 100 milisekund.2:15Tutaj kolejka procesów ma charakter cykliczny.2:21Zakładam, że państwo już mnie słyszą teraz, tak?2:23Tak na pewności się pytam. Dobrze, dziękuję.2:27No a na zmianę dostają po kwancie czasu, czyli tak, proces dostaje kwant czasu.2:34Skroci się kwant czasu, to system operacyjny wyrzuca go z tej kolejki,2:38z tego procesora, a proces wraca do kolejki.2:42No tam w międzyczasie tam inne mogły się pojawić, prawda?2:45Nowe, być inne już w tej kolejce, na końcu tej kolejki się ustawia i tak długo.2:52Tu należy przy tym mówić o przełączaniu kontekstu,2:55czyli to jest ten czas.2:57Związany, o czym mówiłem wcześniej, gdy jest przepisywana informacja z rejestrów procesora3:05do bloku kontrolnego procesu, gdy on jest wyrzucany z procesora,3:10a z kolei ten proces, który dostaje teraz kwant czasu, no to z jego bloku kontrolnego3:14trzeba te informacje wpisać do rejestru procesoru.3:19I czas tego przełączania kontekstu to jest tego rzędu.3:23Jedna do 100 mikrosekund.3:25Oczywiście, jak to się często odbywa, no to też zajmuje pewien czas,3:31no to ta wydajność systemu zależy od tego, no bo w tym czasie procesor nie wykonuje rozkazów żadnych3:39zleconych do wykonania, tylko czeka, aż to, to przepisywanie informacji zostanie zrealizowane, prawda?3:49Dobrze. Teraz, żeby dobrze zrozumieć te wszystkie algorytmy,3:53przygotowałem pewien przykład.3:55Iluztrujący te właśnie algorytmy.4:09Rozpatrzmy taką sytuację, że mamy procesy 1, 2 i 3,4:17które kolejno są zgłaszane w odpowiednio w zerowej sekundzie,4:23potem...4:25sekundę później drugi, jeszcze sekundę później proces trzeci.4:31A przewidywany czas wykonywania tych procesów tu wynosi 1000 sekund,4:37tu wynosi 2 sekundy, tu wynosi dla trzeciego 1 sekundę.4:43Bardzo proszę patrzeć na ten przykład zadania i jego rozwiązanie,4:48bo podobne zadanie będzie domowe.4:52No i proponuję w podobny sposób go rozwiązać.4:55Żeby mieć pewność, że jest dobrze rozwiązany.5:00Procesor jest dostępny do przetwarzania od dziesiątej sekundy,5:04bo wcześniej, przez wcześniejsze procesy był jeszcze zajęty.5:10Tu należy przyjąć czas przełączania kontekstu rzędu 100 mikrosekund.5:16Istotne jest to, że przedstawić na wykresie czasy wykonywania procesów5:20i wyznaczyć średni czas przetwarzania tych procesów.5:25Kolejne zadanie.5:30W celu jego rozwiązania proponuję tutaj zrobić dwie osi czasu,5:37bo na nich należy przedstawić informacje,5:40najpierw wejściowe, a potem końcowe i na nich przedstawić rozwiązanie zadania.5:47Tutaj czasy zgłoszenia procesów widzimy na tej osi czasu,5:52a w zerowej.5:55B w pierwszej, C w drugiej sekundzie zgłoszonej.5:59Tu mamy, że tu jest oś czasu procesora.6:04Tu 10 sekund to jest to, gdy jeszcze ten niebieski czas,6:09to jest czas, gdy procesor jeszcze poprzednie procesy przetwarza,6:16jest wolny od dziesiątej sekundy.6:17No i w tym...6:25Ten algorytm widzi, to jest algorytm systemu operacyjnego,6:33widzi trzy procesy w kolejce.6:37No i zgodnie z algorytmem FCFS,6:40najpierw czas dostanie proces A na tyle, ile chce,6:45czyli na tysiąc sekund, czyli tu widzimy koniec tego procesu,6:49to będzie, on był zgłoszony w zerowej sekundzie,6:53koniec to jest 1010,6:55no bo tu jest tysiąc,6:57plus ta jedna dziesięciotysięczna,7:01to jest ten czas przełączania kontekstu tutaj zaznaczony.7:05To będzie koniec procesu A.7:06Teraz, no znowu, doliczamy czas przełączania kontekstu7:09i proces B kolejny,7:12no to on zakończy się 1012.0022 sekundy7:17i proces C, którego koniec będzie 10137:21i widzimy tutaj tak dalej.7:25Dobrze.7:31Czas cyklu przetwarzania.7:33No to jak wspomniałem, to jest ważne.7:37Czas zakończenia danego procesu minus czas jego zgłoszenia.7:42Zauważmy, że w tym czasie są,7:45no w tym wypadku jest pewny przedział,7:49gdy proces czeka w kolejce, prawda?7:53Do 10.7:55Do 10 sekund i 001, prawda?7:57On czeka tu w kolejce.8:02Czyli ten czas przetwarzania obejmuje czas,8:04gdy proces na coś czeka8:06i proces, gdy on się wykonuje.8:09Podobnie dla procesu tego drugiego8:12i dla procesu tego trzeciego C.8:17No i średni czas przetwarzania do policzenia średnim.8:21To jest sprawa oczywista, tak?8:22Tu widzimy 1010.8:25No to właśnie taki przykład, jakby tu było,8:30jakbyśmy nie stosowali współczesnego systemu Unix,8:34tylko takiego archaicznego,8:38w którym chcemy wykonać polecenie część dyrektory8:41i czekamy na niego 1000 sekund,8:44żeby dostać odpowiedź o jakiś wcześniejszym udługie.8:48Popatrzmy, jak to by wyglądało w przypadku algorytmu SJF.8:54Też mamy,8:55mamy w tych momentach czasu zgłaszane procesy.9:00Też no te 10...9:03Bardzo marnąłem to.9:14Dobrze.9:19Wiesz, to...9:20Więc mamy tak.9:21No znowu czas przełączania kontekstu,9:24ale teraz sprawdzane jest,9:25że kolejka, który jest najkrótszy,9:26najkrótszy jest C,9:27czyli najpierw on dostanie czas procesora.9:31No to odpowiednio jego koniec będzie 1101.9:35Potem mamy proces, kolejny proces,9:38który teraz w kolejce są tylko dwa.9:42Państwo sobie zapiszą.9:46System operacyjny sprawdza kolejkę9:48za każdym razem, gdy zwalniany jest proces.9:53System operacyjny sprawdza kolejkę procesów czekających, procesów gotowych,10:02zawsze wtedy, gdy zwalniany jest procesor.10:07No bo on wtedy musi wybrać, który to następny ma być procesor, prawda?10:12No a teraz w tej chwili kolejnej, tutaj już najkrótszy jest proces A,10:19bo tylko jeden jest, no i on będzie na końcu.10:22I zauważmy, że...10:23Jeśli policzymy sobie średni czas przetwarzania,10:29też pamiętajmy, że czas cyklu przetwarzania kompletnego procesu10:34to jest czas zakończenia minus jego utworzenia, prawda?10:40I w tym czasie też proces, no pewien przedział czasu czeka.10:44I tu mamy średni 344.10:48Czyli widzimy w porównaniu z poprzednim, to jest prawie 1 trzecia.10:53Widzimy, że czas cyklu jest prostu kilkukrotny.10:56Widzimy efektywność tego algorytmu, gdyby go można było zastosować.11:00A teraz popatrzmy na trzeci algorytm rotacyjny.11:08Tu założenie jest, że kwant czasu, żeby taki plan uwzględniany.11:11I gdybyvele Danklya powróci pan w ten sposób,11:13to należy sobie zgłaszać, że אם kwant czasu jest co casually więcej,11:15niż sektor w jakiej okazuje behavioralnie,11:17to zmniejszy się jegoacer atom jest kwant 42вин.11:20To już nie ma takiego brandu.11:21Czyli yety settling Black np. AP.11:22Pani ponieważ my, dzieci, te diecie jest albo jeden ani awaken, matière z jakiego oleh infrastruktury,11:22My też, tak bardzo proszę.11:23No, wiemy, że tak naprawdę te kwanty są dużo krótsze, ale tutaj dla celów takich sprawdzających kwant czasu jest przyjęty jedna sekunda.11:35Mamy też trzy procesy, tak jak poprzednio.11:39Czasy zgłoszenia, czyli tu są te dwie osie czasu.11:47No i tak, będzie wyglądało to tak.11:50W tej dziesiątej sekundzie następuje sprawdzenie kolejki.11:55Mamy i w tym wypadku ten, który był pierwszy zgłoszony rzeczywiście zostaje czas procesora, ale tylko na ten jeden kwant, czyli tylko na jedną sekundę.12:08Teraz zostaje A wyrzucony i sprawdzana jest kolejka.12:14I tu co jest istotne?12:15Państwo na to zwrócą uwagę.12:17Że gdy sprawdzana jest ta kolejka.12:20To tutaj kolejny proces, to jest B w tej kolejce.12:30Ale dalej, jak będziemy patrzeć, to przecież mogą nowe procesy się pojawiać.12:35Trzeba te nowe procesy w tej kolejce uwzględnić, żeby prawidłowo rozwiązać te procesy, ale na razie wrócimy do tego zadania.12:42Potem dostaje kwant czasu proces B, proces C, potem znowu proces A, proces B, a teraz już został tylko A.12:50To A tylko będzie dostawał kwantę czasu.12:53Ale proszę zwrócić uwagę, że po każdym kwancie czasu proces A jest wyrzucany z procesora, sprawdzana jest przez system operacyjny kolejka i potem proces A wraca do procesora, bo tylko jeden jest, prawda?13:10Ale system operacyjny nie wie, czy jakiś inny proces na przykład tutaj się nie pojawił w tej kolejce.13:20Czyli tu chcę zwrócić uwagę, żeby prawidłowo rozwiązać to zadanie, należy za każdym razem sprawdzać tą kolejkę procesu.13:32I tutaj popatrzmy, jaki będzie wynik.13:37No tutaj widzimy, że w przypadku procesu C to będzie 13.003.13:43No tutaj będzie 15.005, bo musimy doliczyć te wszystkie przedziały czasu.13:50Na przełączanie kontekstu, prawda?13:53No i tutaj jak policzymy te wszystkie czasy na przełączanie kontekstu, to wychodzi, że koniec procesu A to będzie 1013.003.14:04Ta jedynka, jedna dziesiąta, no to już z przeliczenia tej pewnej liczby tych przedziałów przełączania kontekstu.14:15Czyli widzimy, że...14:18Zaczyna ta...14:20Liczba przełączania kontekstu tych przedziałów wpływać na ostateczny wynik, prawda?14:28No bo one się doliczają do czasu.14:32I tutaj widzimy, jak będzie w tym wypadku wyglądał wynik.14:38Średni czas przetwarzania widzimy 346 sekund.14:44Dla porównania.14:46To jest nieco gorszy od tego.14:48Tu było 344.14:50Ale proszę zauważyć, jest istotnie.14:54No prawie tak samo dobry jak ten SJF.14:58W porównaniu z tym pierwszym SJF, prawda?15:03Było 1010.15:06Ale co jest istotne w przypadku algorytmu rotacyjnego?15:11Nie ma potrzeby szacowania przyszłego czasu procesora danego procesu.15:18Nie ma potrzeby szacowania przyszłego czasu procesora danego procesu.15:19To jest prawie równie efektywny jak SJF, ale wykonywalny.15:26Czyli dlatego też on jest praktycznie we wszystkich współczesnych systemach operacyjnych.15:32Właśnie ta idea jest stosowana.15:35Oczywiście ona jest bardziej...15:38Stosowane obecnie algorytmy są bardziej złożone, ale opierają się na tej właśnie idei.15:44Tu na jedną rzecz jeszcze chcę zwrócić uwagę Państwu.15:49O czym wspominałem, ale należy na to zwrócić uwagę.15:52Że tutaj, żeby dobrze rozwiązać tego zadanie, tego typu.15:56Gdy jeszcze tu przychodzą w trakcie pewne procesy.16:02Trzeba za każdym razem, tak jak to robi system operacyjny, sprawdzać tą kolejkę.16:08I tu widzimy przykładowo w 12 sekundzie.16:14Czyli 12 to będzie tak. Tu była, a to była 11 jest troszkę.16:18Tu w trakcie procesu B.16:23W trakcie procesu B jak wygląda kolejka?16:29No proces C wszedł w drugiej sekundzie, prawda?16:34A proces A wrócił do kolejki, tu już zaznaczyłem.16:38On wrócił do kolejki 10 sekund plus ten czas przełączania kontekstu.16:48Ale gdyby przykładowo tu w trakcie procesu A.16:54Tu pojawił się jakiś inny proces.17:00No to byłaby sprawa do D jakiś na przykład.17:04No to byłaby sprawa.17:06Gdyby on się pojawił tutaj w trakcie procesu A.17:09No to byłby w kolejce przed procesem A.17:12A gdyby się trafił w kolejce tutaj.17:16W trakcie procesu tego kwantum dla procesu B.17:20No to byłby po procesie A w tej kolejce.17:23Czyli tutaj ta kolejka mogłaby się zmienić.17:27Prawda?17:28Jeśli rozpatrujemy ten moment 12 sekund.17:31I praktycznie każdy moment jeszcze raz przełączania kontekstu.17:36Gdy pewien procesor jest zwalniany.17:41Przez wyrzucany jakiś proces z niego.17:44Ta kolejka jest sprawdzana przez system operacyjny.17:48Na to proszę zwrócić uwagę.17:49To należy podkreślić.17:51Zwłaszcza przy rozwiązywaniu tego zadania.17:55Które za chwilę Państwu przekażę.17:59I które będzie zadaniem do domu.18:03No i właśnie teraz proponuję to zadanie.18:06Przedstawię to zadanie.18:08Zadanie domowe.18:10Tutaj proszę popatrzeć.18:13No dzisiaj to jest wykład piąty i szósty.18:17Będzie jeszcze jedno zadanie.18:19Ale to będzie takie pierwsze.18:21Mamy tak.18:22W chwili zerowej zostaje zgłoszone.18:26Zadanie już wykonać jak zgłoszony proces.18:28I utworzony proces jeden.18:3015 sekund później utworzony proces dwa.18:33I 28 sekund później po procesie dwa proces trzy.18:39Przewidywany czas wykonywania procesu to będzie.18:42Pierwszego 45 sekund.18:44Procesu drugiego 60.18:46Procesu trzeciego 23 sekundy.18:52Procesor jest dostępny do przetwarzania tych procesów od chwili 25 sekundy.18:57I czas przełączania kontekstu proszę teraz pominąć.19:01Żeby ułatwić Państwu rozwiązywanie tego zadania.19:06Czyli to jest uproszczenie.19:09I podobnie jak w tym przykładowym zadaniu.19:12Policzyć średni czas przetwarzania tych procesów.19:15Dla tych algorytmów FCFS.19:17SJF bez wywłaszczania.19:20I algorytmu rotacyjnego przy kwancie czasu.19:23Też żeby łatwiej było liczyć.19:25Tu jest 10 sekund.19:31Tu należy podać sposób rozwiązania i uzasadnienie.19:34I proszę również na wykresie czasu.19:36Zaznaczyć okres wykonywania tych procesów.19:39Czyli bardzo proszę.19:41Tak jak w tym przykładzie tu przedstawianym.19:46Rysować te osie czasu.19:49I zwrócić uwagę na pokazanie na tej osie czasu procesora.19:56Kiedy który proces dostał czas procesora.20:03Na co chcę zwrócić uwagę.20:06Ponieważ na dziennych już tego typu zadania było rozwiązywane.20:10Żeby uniknąć ewentualnych błędów.20:12Proponuję tak.20:13Żeby była ta sama skala tych dwóch osi czasu.20:16Czyli jeśli będzie na przykład proces C.20:20Który się wykonuje w trakcie wykonywania B.20:22To żeby to było widać.20:24Bo wtedy Państwo się nie pomylą.20:26Istotne jest tworzenie tej kolejki.20:29Zaznaczenie tej kolejki w odpowiednich momentach czasu.20:32Czyli jeszcze raz.20:34Kolejka jest sprawdzana.20:38Przez system operacyjny.20:40Zawsze w tym momencie.20:42Gdy zwalniany jest procesor.20:45No tu zwalniany jest procesor.20:47Tu widzimy to tutaj.20:49Gdy z procesora wyrzucany jest proces A.20:52Bo skończył mu się kwant czasu.20:54Potem B i tak dalej.20:56Następna rzecz.20:58Którą też proszę sobie zanotować.21:01Jeśli proces nie wykorzysta całego kwantu czasu.21:05No to w tym momencie właśnie jest uruchamiana kolejka.21:09Jest sprawdzany.21:11Także nie czeka do nowego kwantu czasu.21:13System operacyjny.21:15Tylko od razu wykorzystuje czas procesora.21:17Stara się wykorzystać.21:19Czyli proszę w rozwiązaniu.21:21Nie zrobić takich sytuacji.21:23Ponieważ jeszcze kwant czasu.21:25Kwant czasu jest dłuższy od momentu.21:27Gdy proces został wyrzucony.21:29No to procesor się w tym czasie.21:31Nic nie robi. Nudzi.21:33To nie tak.21:35To byłoby bez sensu.21:37Czyli od razu wtedy.21:38Jeszcze raz.21:39Za każdym razem.21:40Gdy zwalniany jest procesor.21:42System operacyjny sprawdza kolejkę.21:47Czy są jakieś pytania do tego zagadnienia.21:51I do tego typu zadania.21:52Bardzo proszę.22:00No proszę ewentualnie sobie jeszcze raz.22:02No w domu.22:03Korzystając też z nagrania.22:06To popatrzeć na te zagadnienia.22:08Tak jak ja je tutaj przedstawiałem.22:10Ja oczywiście treść tego zadania.22:14Roześlę Państwu.22:16W mailu.22:17Po zakończeniu wykładu.22:23To właśnie wynikające z niesprawdzania tej kolejki.22:26To jest najczęstszy punkt.22:29To mogę jeszcze podać tutaj.22:31W przypadku algorytmu SJTF.22:33Jak mówię dla dziennych.22:35To było tak.22:36Że.22:37Ktoś zaproponował.22:39Że najkrótszy proces.22:42Będzie dostał.22:45No w treści zadania.22:47Najkrótszy proces tam był.22:49Tylko się zaczął.22:51Na przykład tutaj.22:53W tym momencie.22:55No a ktoś stwierdził.22:56Że ponieważ jest najkrótszy.22:57No to należy mu czas przyzielić tutaj.22:59No to wyobraźmy sobie.23:01Skąd system operacyjny.23:03Ma w tym momencie wiedzieć.23:04Jakiś proces krótki przyjdzie.23:05W przyszłości.23:06Czyli jak.23:07Czyli wynikające z tego co.23:09Tutaj kilka razy już powtarzam.23:12System sprawdza kolejkę.23:14Zawsze wtedy.23:15Gdy zwalniany jest czas procesor.23:19Gdy zwalniany jest procesor.23:20Prawda.23:21No i.23:22Trzeba wybrać kolejny proces.23:23Do wykonania.23:24Prawda.23:25Także to były najczęstsze błędy.23:27I dotyczyły.23:28I algorytmu.23:29SJF.23:30I.23:31No w tym pierwszym.23:32To trudno się pomylić.23:34Bo jest taki.23:35Bo jest bardzo prościutki.23:36Ten algorytm.23:37Ten FCFS.23:38Prawda.23:39Ale tu chodzi mi o to.23:40Żeby państwo.23:41Dobrze rozumieli.23:42Ideę tych algorytmów.23:43No i zalety.23:44Tego rotacyjnego.23:45W porównaniu.23:46Z tymi ideami.23:47Które czasami.23:48Były wcześniej stosowane.23:49Czy próbowane.23:50Były do.23:51Zastosowania.23:52Czy może.23:53Jeszcze.23:54Jakieś inne.23:55Pytanie.23:56Ze strony.23:57Państwa.23:58Bardzo proszę.23:59Jeśli nie.24:00To ja.24:01Zamknę.24:02Te.24:03Te.24:04Dodatkowe.24:05Dodatkowe pliki.24:06I.24:07Przejdziemy.24:08Teraz.24:09Do tego.24:10Jak.24:11Wygląda.24:12Szeregowanie.24:13Procesów.24:14W.24:15Systemie.24:16Unixowym.24:17Czyli.24:18Jeszcze.24:19Raz.24:20To.24:21Co.24:22Do tej pory.24:23Przedstawiałem.24:24To.24:25Były.24:26Pewne.24:27Idea.24:28Różnych.24:29Algorytmów.24:30A.24:31Teraz.24:32Mówimy.24:33Jak.24:34To.24:35Jest.24:36Pierwsze.24:37System.24:38To.24:39Jest.24:40System.24:41Z.24:42Podziałem.24:43Czasu.24:44Procesora.24:45Czy.24:46Mówiłem.24:47Wiele.24:48Czyli.24:49Jądro.24:50Systemu.24:51Operacyjnego.24:52Przydziela.24:53Procesor.24:54Procesowi.24:55Gotowemu.24:56Na.24:57Jeden.24:58Kwant.24:59Czasu.25:00Co.25:01Naj.25:02Państwo.25:03No.25:04To.25:05Zostanie.25:06Wyrzucony.25:07Jeśli.25:08To.25:09Będzie.25:10Login.25:11Shell.25:12Który.25:13Wywoła.25:14Potomka.25:15I.25:16Się.25:17Uśpi.25:18No.25:19To.25:20Będzie.25:21Wyrzucony.25:22Z.25:23Procesora.25:24Czyli.25:25Może.25:26Zdarzyć.25:27Że.25:28Proces.25:29Nie.25:30Wykorzysta.25:31Całego.25:32Po.25:33Upływie.25:34Tego.25:35Czasu.25:36Wy.25:37Wy.25:38Właszcza.25:39Proces.25:40Czyli.25:41Po.25:42Tym.25:43Kwancie.25:44I.25:45Przydziela.25:46Procesor.25:47Procesowi.25:48Następnym.25:49Kolejce.25:50W.25:51Kolejce.25:52Przy czym.25:53Za.25:54Każdym.25:55Razem.25:56Wybiera.25:57Proces.25:58Który.25:59Jest.26:00W.26:01Następna.26:02Rzecz.26:03Te.26:04Priorytety.26:05Nie.26:06Są.26:07Stałe.26:08One.26:09Są.26:10Przeliczane.26:11Dlaczego.26:12No.26:13Z.26:14Tego.26:15Powodu.26:16Że.26:17No.26:18Wyobraźmy.26:19So.26:20Sytuację.26:21Taką.26:22Że.26:23Jest.26:24Proces.26:25Który.26:26Ma.26:27Niski.26:28Priorytet.26:29I.26:30Bardzo.26:31Długo.26:32Czekała.26:33Na.26:34Dostęp.26:35Do.26:36Procesora.26:37A.26:38Może.26:39I.26:40Nigdy.26:41Prawda.26:42No.26:43Bo.26:44Ciągle.26:45Mogły.26:46Być.26:47O.26:48Wyższych.26:49Priorytetach.26:50Więc.26:51W związku.26:52Z.26:53Tym.26:54Tutaj.26:55Te.26:56Priorytety.26:57Procesów.26:58Są.26:59Modyfikowane.27:00No.27:01Takie.27:02Zjawisko.27:03Gdy.27:04Proces.27:05Bardzo.27:06Długo.27:07Czeka.27:08Na.27:09Pewien.27:10Zasób.27:11W.27:12Szczególności.27:13Dostęp.27:14Do.27:15Procesora.27:16Nazywamy.27:17Sytuacją.27:18Głodzenia.27:19Procesu.27:20Żeby.27:21Uniknąć.27:22Tego.27:23Głodzenia.27:24Procesu.27:25No.27:26Po.27:27W.27:28By.27:29Przeczuł.27:30Te.27:31Priorytety.27:32Mają.27:33Wartości.27:34Całkowito.27:35Liczbowe.27:36Czyli.27:37Wywłaszczony.27:38Proces.27:39Jest.27:40Umieszczany.27:41W.27:42Jednej.27:43Z.27:44Kolejek.27:45Priorytetowych.27:46Czyli.27:47Mamy.27:48Różne.27:49Poziomy.27:50Priorytetów.27:51Jak.27:52To.27:53Rozumieć.27:54Mamy.27:55Różne.27:56Poziomy.27:57Priorytetów.27:58Czyli mamy tak, podział czasu procesora na kwanty, algorytm rotacyjny, algorytm priorytetowy, które są jeszcze modyfikowane w czasie28:15i mamy w rezultacie kolejki priorytetowe, kolejki na różnych poziomach priorytetów, których może być jeden lub więcej procesorów, żadnego.28:28I teraz, ten algorytm szeregowania można przedstawić formalnie w postaci takich trzech punktów.28:37Pierwszy, czyli tak, mamy sytuację, że procesor został zwolniony przez jakiś proces, ten powód może być różny.28:51W tym momencie uruchamiany jest algorytm i wtedy algorytm ten sprawdza,28:58czy coś tu się dzieje niedobrego.29:20Dobrze, poprawiło się.29:22Czyli sprawdza kolejkę procesów gotowych29:26i wybiera,29:28kto ma spośród nich ten, który ma najwyższy priorytet.29:32Może być sytuacja, że jest kilka procesów o tym samym priorytecie.29:37To wtedy spośród nich wybiera ten, który najdłużej czeka w stanie gotowym.29:44A może być sytuacja, że w danym momencie nie ma procesów gotowych do wykonania.29:50No to czeka do następnego przerwania,29:53przecież kolejnego taktu zegara i znowu sprawdza to i tak dalej.29:58Jeszcze weźmy pod uwagę, że ta kolejka cały czas może się zmieniać,30:02bo nowe procesy mogą przychodzić.30:04Mogą dochodzić do niej procesy uśpione z różnych powodów.30:11Mówimy o systemach komercyjnych, czyli takich, których nawet jest nie jeden,30:17ale wielu użytkowników w tym czasie działających.30:20No i są jeszcze procesy systemobracyjne.30:23Zauważmy, że ta sytuacja jest dość złożona,30:26więc tutaj można powiedzieć, że przewidzenie czasu,30:30ile dany proces czy dane polecenie będzie wykonywane jest praktycznie niemożliwe.30:34Bo sytuacja jest dynamiczna cały czas.30:39No a teraz przejdźmy do tych priorytetów.30:42Tutaj muszę Państwu troszkę więcej powiedzieć30:48o zagadnieniach związanych z wykonywaniem procesów w trybie użytkownym.30:56Użytkownika i wykonywanych w trybie jądra.31:02Tutaj mamy tak, no powiedzmy, że proces wykonuje się w trybie użytkownika31:07i ten proces zażyczył sobie, żeby system operacyjny wykonał dla niego pewne działanie.31:14No przykładowo została użyta pewna funkcja systemowa31:18lub też wystąpiło przerwanie z jakichś względów.31:22Wtedy ten, jak wspominałem wcześniej, ten proces przechodzi31:26z wykonywania w trybie użytkownika do wykonywania w trybie jądra.31:31No i w trybie jądra następuje obsługa tej funkcji systemowej,31:38która wymaga dodatkowych działań ze strony systemu operacyjnego.31:42I przykładowo, jeśli to będzie na przykład żądanie operacji dyskowej,31:48no zapisu czy odczytu, no to w tym celu system musi użyć31:56odpowiedniego dostępu do tego dysku, żeby tę operację wykonać.32:04Wtedy następuje uśpienie.32:06To się wykonuje ciągle jako ten fragment kodu systemu operacyjnego,32:11wykonuje się dalej jako ten sam proces.32:14W ramach tego procesu wykonuje się fragment jądra.32:19Wtedy następuje uśpienie, czyli wyrzucenie tego procesu32:25do systemu procesora.32:29To uśpienie trwa tak długo, aż ta operacja, no przykładowo,32:33ta operacja dyskowa zostanie wykonana.32:36Gdy ona zostanie wykonana, to nastąpi budzenie procesu.32:40On jeszcze jest w trybie jądra gotowy do wykonania.32:46No ale żeby wrócił do trybu użytkownika, to znowu musi dostać czas procesora32:53i wraca do procesu.32:55Do dalszego wykonywania w trybie jądra.32:58I istotne jest teraz to, że w tym czasie, gdy proces wykonywany jest w trybie jądra,33:06on ma odpowiednio wysoki priorytet.33:10Bo trzeba zapewnić, żeby to korzystanie z systemu operacyjnego,33:15z jądra systemu operacyjnego było możliwe jak najbardziej efektywnie.33:19I te priorytety są właśnie różne w zależności od powodu,33:23dla którego on,33:25jest wykonywany.33:28Czyli powodu, dla którego on wykonywany jest w tym trybie jądra33:31i dlatego tutaj został uśpiony tutaj.33:38Tu jeszcze na jedną rzecz chcę zwrócić uwagę.33:41W trakcie wykonywania procesu w trybie jądra,33:45mogą, nie muszą, ale w pewnych sytuacjach są przetwarzane33:50wewnętrzne struktury jądra systemu operacyjnego.33:55I mogłoby się zdarzyć, że w tym czasie, gdy one są przetwarzane,34:00gdy proces wykonuje się w trybie jądra,34:03pojawiłoby się pewne przerwanie.34:08Gdyby w tym momencie obsłużyć to przerwanie i przerwać wykonywanie procesu,34:15te wewnętrzne struktury mogłyby ulec uszkodzeniu.34:20Z tego względu,34:22gdy są,34:24są przetwarzane wewnętrzne struktury jądra,34:29przerwanie nie jest obsługiwane.34:33A obsłużone zostanie dopiero wtedy,34:36gdy proces powróci do wykonywania w trybie użytkownika.34:40Gdyby przerwanie wystąpiło w trakcie wykonywania procesu w trybie użytkownika,34:46no to jest od razu obsłużone bez czekania.34:50No ale także,34:51gdy proces wykonuje się w trybie jądra,34:53nie są przetwarzane wewnętrzne struktury,34:56również przerwanie może być od razu obsłużone,34:59bo nie ma niebezpieczeństwa,35:01że nastąpi zapaść systemu operacyjnego.35:05Czyli jeszcze raz,35:06gdyby były przetwarzane wewnętrzne struktury systemu operacyjnego35:13i nastąpiło przerwanie i to przerwanie spowodowało przerwanie wykonywania tego procesu,35:21w tym momencie,35:23uszkodzone byłyby wewnętrzne struktury systemu35:28i nastąpiłaby zapaść całego systemu.35:31Nie można do tego dopuścić.35:36Dobrze.35:37No to teraz przejdziemy do właśnie,35:40tutaj bardzo proszę zwrócić uwagę,35:42bo to też było podobne zadanie,35:45które dla Państwa też będzie dotyczące tych priorytetów,35:50które będę chciał przedstawić.35:53Proszę zwrócić uwagę,35:54jak to wyglądają te priorytety.35:58Mamy, czyli tak,35:59mamy procesy,36:00które wykonują się w danym momencie w trybie użytkownika i w trybie jądra.36:04Tu widzimy te priorytety poziomu użytkownika,36:07tu są nisko,36:09poniżej tak zwanej wartości progowej.36:13Natomiast priorytety poziomu jądra są odpowiednio wysokie.36:18I co jest istotne?36:19Proszę zobaczyć.36:20Tu przykład.36:22Oczekiwanie na zakończenie potomka.36:26Czyli tu widzą Państwo wymienione powody,36:29dla którego proces się wykonuje w trybie jądra.36:33Oczekiwanie na zakończenie potomka.36:36Oczekiwanie na wyjście z terminalu.36:38To są przykłady,36:39bo tak naprawdę ich jest więcej.36:41Ale tu chciałem na tych przykładach zilustrować właśnie te priorytety,36:49różne poziomy priorytetów i kolejki priorytetowe.36:51Oczekiwanie na wyjście z terminalu.36:54I tu co jest istotne?36:56Te przykłady,36:58zakończenie potomka,37:00wyjście z terminala,37:02wejście z terminala,37:04to są przykłady wykonywania procesu w trybie jądra,37:10które są przerywalne.37:12Czyli takie, gdzie ich wykonywanie można przerwać,37:18ponieważ nie są przetrwane.37:20Nie są przetwarzane wewnętrzne struktury systemu operacyjnego.37:26Państwo próbują to wszystko zanotować,37:28bo tego czasami nie znajdzie się w literaturze.37:31A chcę, żeby państwo jednak dobrze te zagadnienia rozumieli.37:37Czyli mamy poziomy priorytetów,37:41w trakcie których wykonywania w trybie jądra procesu,37:50nie są przetwarzane wewnętrzne struktury systemu operacyjnego37:55i wtedy przerwanie może być obsłużone bez zagrożenia dla systemu.38:01Natomiast tu wyżej mamy przykłady38:05poziomów priorytetów nieprzerywalnych.38:09Czyli takich, że wtedy są przetwarzane wewnętrzne struktury38:14i mogłoby to,38:16gdyby wtedy przerwać wykonywanie procesów,38:19w trakcie tego mogłoby nastąpić uszkodzenie wewnętrznej struktury38:24systemu operacyjnego i jego zapaść.38:27Tu widzimy takie oczekiwanie na i-węzeł,38:31oczekiwanie na bufor,38:34oczekiwanie na dyskową operację wejścia-wyjścia, proces wymiany.38:39Kolejność tutaj ustawienia tych priorytetów nie jest przypadkowa.38:45Zauważmy,38:47żeby wykonać przykładowo operację dyskową zapisu,38:52to najpierw musi system operacyjny zajrzeć do i-węzła,38:59tego pliku, który ma zapisać.39:03Tam znajdzie informacje adresowe inne.39:08Teraz musi skorzystać z bufora.39:15Tu na jedną rzecz chcę zwrócić uwagę.39:18Operacje dyskowe są w systemach UNIX-owych buforowane w tym sensie,39:24że gdy system operacyjny na życzenie procesu ma zapisać pewne informacje,39:33te informacje są zapisywane z pewnego bufora.39:37Czyli najpierw proces musi generować informacje wyjściowe do tego zapisu,39:45przykładowo w pliku,39:47do takiego bufora,39:49gdzie tworzone są bloki dyskowe39:52i dopiero całe bloki dyskowe są zapisywane na dysk.40:00Ten zapis na dysk jest obsługiwany przez taki proces daemon.40:07Napiszę to po angielsku.40:13No ja przeliteruję po polsku.40:16SYNC od synchronizacji, prawda?40:19SYNC, no SYNC po prostu dokładnie mówiąc,40:22a zapisując SYNC,40:24który przykładowo, no na przykład co 3 sekundy na przykład,40:28dokonuje takiego zapisu bloków z bufora na dysk.40:36Przy czym to jest istotne,40:39że ten bufor oczywiście mieści pewną liczbę,40:42bloków, prawda?40:46Dopiero gdy przyjdzie nowy blok,40:49no to ten najstarszy będzie usuwany,40:52a nowy blok będzie zapisywany w tym buforze.40:55Czyli tam w tym buforze no mieści się określona liczba bloków,40:59no i te najstarsze są w przypadku zapisu informacji do bufora,41:03no to są wymieniane na te najnowsze.41:06Ale dalej, co tutaj jest istotne?41:11Istotne jest to, że przy odczycie z kolei,41:14system operacyjny najpierw próbuje odczytać te bloki potrzebne,41:21o które jest prośba z tego bufora.41:25Nie z dysku, z bufora.41:29Jeśli one są w tym buforze,41:32to operacji dyskowej w ogóle nie trzeba wykonać.41:37Ale dopiero gdy ich nie ma,41:39no to wtedy trzeba wykonać operację dyskową,41:42ale wtedy po wyniku tej operacji dyskowej41:46zostaje to zapisane do tego bufora dopiero,41:50a dopiero potem z tego bufora,41:54no na życzenie procesu, proces dostanie tą informację.41:58Także tak wygląda operacja czytania i operacja zapisu42:03z wykorzystaniem tego bufora,42:07który przyśpiesza wykonywanie operacji42:14związanych z odczytem lub zapisem informacji na dysk, prawda?42:19No bo w przypadku odczytów w pewnych sytuacjach może się okazać,42:23że samej operacji nie trzeba wykonać w ogóle dyskowej.42:27Czyli tak.42:31No a nawet jeśli trzeba wykonać,42:34to i tak najpierw ta informacja musi być wpisana,42:36a dopiero z bufora będzie przekazana do procesu,42:43gdy proces zażyczył sobie operacji odczytu.42:46Czyli tak, widzimy, że żeby dokonać przykładowo tej operacji odczytu,42:51to najpierw musi system operacyjny,42:54no i ten proces w trakcie wykonywania,42:59to jest wykonywane w trybie jądra, zauważmy.43:03W ramach kodu systemu operacyjnego,43:05czyli najpierw jest istotna informacja o I-węźle,43:11który to jest I-węzeł, potem dostęp do bufora,43:15a potem też, jeśli trzeba, bo może się okazać,43:18że już operacji dyskowej nie trzeba wykonać,43:20a jeśli trzeba, no to trzeba wykonać tą operację dyskową.43:24Czyli tu jest najniższy spośród tych priorytetów, tu jest najwyższy.43:30No jeszcze wyższy priorytet ma proces wymiany dotyczący wymiany procesu,43:34między, przepisaniem można powiedzieć,43:39między pamięcią operacyjną a dyskiem.43:42Gdyby chcieć wykonać proces, którego nie ma pamięci operacyjnej,43:47no to trzeba go najpierw ściągnąć,43:49lub jego odpowiednie części ściągnąć z pamięci operacyjnej.43:53Przepraszam, z obszaru słup na dysku do pamięci operacyjnej.43:59Ale o tym będę mówił, gdy przejdziemy do zarządzania pamięcią.44:04Czyli taka jest kolejność tutaj nieprzypadkowa,44:08ale istotna tych priorytetów.44:11I zauważmy tutaj, to jest narysowane z prawej strony,44:15że na każdym poziomie priorytetów może być pewna kolejka procesu.44:24Tu przykładowo jest jeden, tu są trzy, tu są dwa, tu jeden, tu są trzy i tak dalej.44:31Czyli jeśli tu jest jeden, tu są trzy, tu są dwa, tu jeden, tu są trzy i tak dalej.44:33Czyli jeśli tutaj wrócimy do tego algorytmu szeregowania.44:40Jeszcze tutaj parę razy, trzeba to prawo przewrócić.44:44Tutaj widzimy, czyli jeśli jest kilka o równym priorytecie,44:52to wybierz proces najdłużej czekający w stanie gotowym.44:57Czyli tak, a tu patrzmy, tu jest ten, który ma najwyższy priorytet.45:01Czyli najpierw sprawdzamy ten, który ma najwyższy priorytet.45:03Która to jest kolejka według poziomu priorytetów.45:07A teraz w ramach tej kolejki, który proces najdłużej czeka.45:11Jeszcze raz.45:17Dobrze, czyli tu widzimy, na każdym priorytecie może być kilka procesów.45:21No i tutaj, gdyby była taka sytuacja, to najpierw czas procesora dostanie ten proces,45:28który ma ten priorytet.45:30Potem dopiero kolejno.45:31Gdy to już takiego nie będzie, to dopiero będzie ta kolejka sprawdzana.45:37I ten, który najdłużej czeka, to jako pierwszy.45:41No i tak dalej.45:43Gdy już takich nie ma, no to tam i tak dalej.45:46Dobrze, a teraz popatrzmy na poziomy użytkownika.45:52Tutaj też mamy różne poziomy tych priorytetów.45:59Przy czym tu jest używana taka zmienna, która przyjmuje wartości od całkowite 0,1 do n.46:11Przy czym im wartość tej zmiennej jest niższa, tym priorytet jest wyższy.46:20Czyli poziom użytkownika 0 to jest ten o najwyższym priorytecie,46:25a ten n to jest o najniższym.46:29Tutaj takie krótkie podsumowanie.46:46Priorytety poziomu jądra przydzielane są procesom,46:51wtedy, gdy wykonują operację SLEEP, przechodzą stan uśpienia.46:56One wykonywane są w trakcie jądra.46:59Tak jak poprzednio pokazywałem.47:01W zależności od przejścia procesu do tego konkretnego stanu.47:05I teraz mamy tak.47:08Procesy poziomu użytkownika z kolei.47:11A te powody przejścia, no to właśnie w zależności od tego, jaki to jest powód przejścia do tego stanu,47:19czyli jak na przykład zażądanie operacji dyskowej,47:23czy wejście z terminala, czy wyjście z terminala,47:27czy zakończenie potomka.47:30No to odpowiednio będzie ten poziom priorytetu, tak jak na tym poprzednim schemacie przedstawiłem.47:35A teraz popatrzmy dokładniej na priorytety poziomu użytkownika.47:39One są nadawane wtedy, gdy proces wraca z trybu jądra do trybu użytkownika.47:44Jak wspomniałem, one są właśnie przeliczane, te priorytety.47:51Tu jest przykład systemu FIVE.47:55To jest tam taki program obsługi zegara, na przykład co jedną sekundę,48:00przelicza zużycie czasu procesora przez wszystkie procesy48:04i wyznacza odpowiednie priorytety w ten sposób, na przykład jak tu podmieniowy.48:10Czyli priorytet, ale tu jest, przez słowo priorytet, to jest tutaj jest,48:15to jest ta zmienna, wartość tej zmiennej, zmienna priorytet.48:22Jak tu niżej lepsza.48:24Niska wartość zmiennej, priorytet.48:27Ten od zera do n, co przedstawiłem poprzednio, to oznacza wysoki priorytet.48:31Czyli najwyższy to jest taki, gdy ta zmienna wynosi zero.48:35A tutaj popatrzmy, priorytet, ta zmienna przeliczana jest w ten sposób.48:40Ostatnie zużycie procesora.48:43Od centra, procesor, unit.48:46Plus pewien priorytet bazowy, to jest ten rozgraniczający procesy,48:51te priorytety trybu jądra.48:53Proces priorytetu trybu użytkownika.48:56Ten priorytet bazowy to właśnie tą wartość progową przyjmuje.49:00Czyli będzie to to zero, prawda?49:05I teraz tak, jeszcze raz podkreślam, że niska wartość tej zmiennej,49:10bo to tutaj kursywą napisane to jest zmienna priorytet,49:14oznacza wysokość, priorytet, wysoki priorytet szeregowania.49:18A jest taka funkcja systemowa nice i też jest polecenie nice.49:23Gdzie należy podać pewną wartość.49:28Pozwala zmienić tej zmiennej priorytet wykonywanego procesu.49:33Czyli pełna postać tego przeliczania.49:39Tu się dodaje wartość, jeszcze tą wartość podaną za pomocą tej zmiennej nice.49:45I teraz tak, administrator może nadawać procesom wartości,49:51wyższe od progowej.49:55Czyli inaczej mówiąc administrator może podwyższać wartości priorytetu procesom.50:01Użytkownicy mogą tylko obniżać.50:05No i tutaj tak można sobie zażartować, dlaczego to polecenie i zmienna nazywa się nice.50:13Bo użytkownicy mają polecenie, mogą użyć polecenia nice.50:19Tutaj to n to podaje się tutaj pewną liczbę.50:23Na przykład z przydziału 1.19, ale to zależy od implementacji systemu operacyjnego.50:27I wiersz polecenia.50:29No od 1 do 19 można podać tą wartość.50:36Ale ja mówię to jako przykład, bo oczywiście w konkretnej implementacji może to być inaczej.50:41Ale Państwo sobie mogą użyć manuala men i tam będzie pewnie polecenie nice.50:48Z opisem.50:50Natomiast ja mówię o co tutaj chodzi.50:52No ale to już jako ciekawostka taką, że użytkownik może używając tej zmiennej nice,51:00tego polecenia nice po prostu obniżyć swój priorytet w rezultacie.51:06No bo jak wyższa zmienna, wartość tej zmiennej to jest niższy priorytet.51:11Czyli dlaczego to jest nice?51:13No jest miły dla koleżeństwa.51:15Bo jak swój priorytet.51:17Przyorytet polecenia obniży.51:19No to inne osoby będą miały, inni użytkownicy ich polecenia szybciej się wykonają po prostu.51:27No ale to jako żart podaję tutaj.51:31Powód dla którego ta zmienna nazywa się nice.51:34No jestem tak miły dla innych, że obniżę priorytet.51:37No przykładowo jeśli komuś nie zależy na szybkim wykonywaniu.51:41I przekonuje jakieś przetwarzanie bazy danych na noc na przykład.51:44Sobie włączył.51:46No to jemu to nie zaszkodzi, że trochę obniży swój priorytet polecenia.51:52A inne bieżąco wykonywane się szybciej wykonają.51:56Procesy potomne dziedziczą wartości nice.52:01Proces nie może zmienić wartości nice innego procesu.52:05To też jest istotne.52:07No to jest ochrona działania użytkowników.52:11Prawda?52:12Że nie może jeden wpływać na nice.52:14Na nice innego.52:17Dobrze.52:20Czy te zagadnienia to w miarę udało się Państwu wszystkim dobrze złapać?52:28To ja tutaj podam przykład zadania domowego.52:38Będzie następne.52:40To jest drugie zadanie na dzisiaj.52:42Na potwierdzenie obecności.52:44Też w ramach tych dwóch dzisiejszych wykładów.52:48Mamy takie procesy gotowe.52:54Mamy jeden proces w trybie użytkownika.52:58Mamy teraz procesy w trybie jądra.53:03Tu już jeden działa w trybie jądra z powodu oczekiwania wejścia z terminala.53:10Czeka tyle czasu.53:12Proces C w trybie jądra też z powodu oczekiwania, ale na dostęp do bufora.53:18Czeka 0,3 sekundy.53:20Proces D w trybie jądra z powodu oczekiwania na dostęp do IWZ-ła.53:24Czeka 0,2 sekundy.53:26W jakiej kolejności uzyskają dostęp do procesorów?53:30Które z tych procesów mogą być przerwane w trakcie wykonywania na przykład w celu obsłużenia prerwania?53:40W celu dobrego odpowiedzenia na to pytania należy po prostu spojrzeć na ten slajd.53:48Tu jest wymieniony algorytm szeregowania. Proszę dokładnie go zczytać.53:58To jest w zasadzie potwierdzenie obecności.54:01Czy Państwo dokładnie przyswoili ten materiał?54:04To jest jedno.54:06A teraz to jest jedno.54:08I korzystać.54:10Z tego tutaj, prawda?54:16Tu akurat takie przykłady wzięte w tym zadaniu, które występują tutaj na tym slajdzie.54:25W zasadzie zadanie na rozumienie tych zasad algorytmu priorytetowego stosowanego w systemach Unix-owych.54:38Czy...54:40Jakieś pytania może do tego zadania?54:42Bardzo proszę.54:46Ja zdaję sobie sprawę, że no ja muszę dość dużo tych informacji naraz Państwu przekazać ze względu na krótki czas na wykłady.54:55I muszą Państwo sobie spokojnie w domu przejrzeć nagranie i jeszcze raz to wszystko przeanalizować.55:02Żeby dobrze przyswoić.55:04Ale temu służą właśnie te zadania.55:06Że bardzo proszę rozwiązywać te zadania.55:08No bo to jest przygotowanie do tego końcowego sprawdzianu z całego kursu.55:15No praktycznie ma charakter egzaminu, prawda?55:18Ale to nazywa się sprawdzianem.55:21Dobrze. Przejdziemy dalej.55:24Przejdziemy teraz do kolejnego zagadnienia.55:28To jest do zagadnienia ładowania systemu operacyjnego.55:36Czyli rozpatrujemy sytuację taką, że mamy system komputerowy, w którym jest system operacyjny, ale gdzieś zapisany na dysku.55:49I włączamy. Włączamy zasilanie.55:54No i żeby system zaczął działać, to system operacyjny musi się znaleźć, czyli być załadowany do pamięci operacyjnej.56:05Czyli tu właśnie omówimy jak to jest realizowane, to ładowanie.56:13I jak ten system operacyjny zaczyna działać.56:17No po angielsku ta operacja ładowania to nosi nazwę bootstrap.56:22Czyli tak. Opowiadałem Państwu wcześniej, gdy mówiłem o partycjach dyskowych.56:30Że na dysku głównym, bo czasami tych dysków może być więcej.56:34Jest pewien obszar pamięci, w którym jest właśnie tak zwany boot area, czyli blok zawierający program służący temu ładowaniu systemu operacyjnego do pamięci.56:49No i właśnie.56:52Zaczyna się to po włączeniu zasilania.56:57Zaczyna się pierwsza rzecz, inicjalizacja i testowanie sprzętu.57:03A następnie właśnie następuje wczytanie do pamięci tego bloku systemowego, który zawiera program.57:17Przecież to jest nazwany blokiem zero z dysku, który zawiera program służący temu ładowaniu systemu operacyjnego.57:27Następnie, gdy to zostanie załadowane do pamięci operacyjnej.57:31To ten program tam zawarty, ładuje jądro systemu operacyjnego z pliku, który jest na dysku.57:40Który ma swoją nazwę.57:43Tu jest przykład takiego pliku w systemie HPUX.57:49Stand Virtual M-UNIX.57:55Czyli następuje ładowanie do pamięci operacyjnej tego bloku.57:59Tego jądra systemu operacyjnego z pliku, który jest na dysku.58:06I następnie, gdy on jest całkiem już załadowany.58:09To następuje przekazanie do pierwszej już instrukcji tego jądra.58:15I on zaczyna się wykonywać.58:20Wykonuje się już program jądra systemu operacyjnego.58:26No i zaczyna od wykrywania i konfiguracji urządzeń.58:31Musi znaleźć główny katalog plików.58:35I przygotowuje środowisko procesu o identyfikatorze zero.58:41Ten program, ten proces wykonuje się jeszcze jako proces w trybie jądra.58:49On tworzy pewne procesy jądra, procesów zarządzania pamięcią.58:54I następuje następnie wywołanie funkcji systemowej po oku.59:01Czyli on się rozwidla ten proces zero.59:05I tworzy proces o identyfikatorze.59:08No swego potomka, czyli proces tworzy o identyfikatorze jeden.59:14Tworzy kontekst poziomu użytkownika.59:16I już przechodzi do wykonywania w trybie użytkownika.59:20Czyli proces PID1.59:23Wykonuje się już, jest to proces systemu operacyjnego,59:26ale wykonuje się w trybie użytkownika.59:30Teraz, ten proces jeden teraz ma, następuje wywołanie dla niego funkcji systemowej x.59:40I wywołuje, to jest ten program, który ma się przez niego wykonywać.59:50To jest właśnie ten SBIN.59:52On jest w katalogu SBIN.59:55A on ma nazwę init.59:58I to jest ten proces, który cały czas działa w systemie operacyjnym.1:00:05I o nim będę dalej teraz mówił.1:00:11Ten proces init, on wczytuje plik init.tab, który jest w katalogu ETC na ogół.1:00:21I rozmnaża odpowiednie procesy.1:00:27On inicjuje wewnętrzne struktury danych jądra systemu operacyjnego.1:00:33Tworzy listy wolnych buforów, i węzłów, kolejek i tak dalej.1:00:41I co jest istotne?1:00:43init.tab.1:00:45init.tab to jest plik, który może być edytowany przez administratora.1:00:50Tam są w szczególności informacje dotyczące możliwości łączenia z różnymi terminalami.1:00:58W jaki sposób łączyć, jak w jaki sposób współpracować, jak przesyłać informacje poprzez łącza do różnych terminali.1:01:09Czyli można, to tak już jako ciekawostka.1:01:12Może administrator na przykład, on bo jest edytowany.1:01:15Można tam napisać, że określone terminale mogą być używane tylko w godzinie.1:01:18Mogą być używane tylko w godzinie na przykład ósmej do szesnastej.1:01:21Gdyby ktoś tego terminala chciał po szesnastej skorzystać, to nie będzie mógł korzystać.1:01:29Czyli ogólnie biorąc to, tam są informacje dotyczące łącz z różnymi terminalami.1:01:37Dobrze.1:01:39Następnie następuje sprawdzenie.1:01:46Wiedzą Państwo, że1:01:48w złożonych systemach może być główny i może być więcej niż jeden z systemów PIC-ów.1:01:54I następuje sprawdzenie tych wszystkich systemów PIC-ów, zarówno głównego jak i pozostałych.1:01:59I tutaj istotne jest takie,1:02:04czasami jest taka potrzeba użycia1:02:07takiego programu File System Check, FSCK.1:02:15Otóż, on nie zawsze musi być uruchamiany.1:02:17jak Państwo sami rozumieją z nazwy, sprawdzania prawidłowości poszczególnych systemów plików.1:02:28File System Check.1:02:32Tutaj istotny jest problem prawidłowego zamknięcia.1:02:36No wtedy, gdy zamykamy system z kolei.1:02:40Prawidłowego zamknięcia systemu plików i wszystkich zamknięcia wszystkich systemów.1:02:47Całego systemu operacyjnego i wszystkich systemów plików.1:02:51Bo taki system plików może być prawidłowo zamknięty, a może być nieprawidłowo zamknięty.1:02:58Kiedy jest system nieprawidłowo zamknięty?1:03:04Gdyby przykładowo1:03:08w trakcie pracy systemu1:03:11nastąpiło wyłączenie zasilania.1:03:14Co się może wtedy stać?1:03:17Wspomniałem już Państwu wcześniej, że operacje dyskowe są buforowane.1:03:22Czyli mamy pewne informacje, które na bieżąco zmieniane są w buforze.1:03:28A tylko co pewien czas są zapisywane na dysk.1:03:35Czyli są przedziały czasu, w którym te informacje mogą być rozbieżne.1:03:41Czyli rozbieżne może być to co jest zapisane na dysku z tym co jest zapisane w tych głównęcznych buforach.1:03:47I gdyby1:03:49w tym momencie właśnie1:03:51rozbieżności tych informacji1:03:53nastąpiło wyłączenie zasilania.1:03:55No to1:03:57te informacje zapisane na dysku mogą być nieprawidłowe.1:04:02I tego szuka file system check.1:04:05Niektóre błędy potrafi znaleźć, niektórych nie.1:04:08Ale istotne jest przede wszystkim1:04:11to, żeby1:04:13prawidłowo zamykać system. O tym będę mówił.1:04:17Ten proces1:04:19init1:04:21jego jednym z głównych zadań jest1:04:24monitorowanie1:04:26i zapewnienie możliwości pracy1:04:29terminali w ramach systemów, podłączonych systemów.1:04:33W tym celu on1:04:35generuje takie procesy,1:04:37czyli tworzy procesy o nazwie gdty.1:04:40To przy poleceniu PS dałem taki przykład Państwu.1:04:45Które1:04:46monitorują konsele i terminale systemu komputerowego.1:04:51To się odbywa zgodnie właśnie z tymi1:04:53deklaracjami w pliku init tab, o którym mówiłem przed chwilą.1:04:58Czyli generuje kolejny proces gdty.1:05:01Daje jednego terminala, drugiego, trzeciego1:05:04i za każdym razem go usypia.1:05:07To usypia, zostanie obudzony.1:05:09Za jakiegoś powodu.1:05:12On również tworzy takie procesy demony.1:05:15To realizujące pewne zadania systemu operacyjnego.1:05:21Teraz popatrzmy na zagadnienie1:05:24otwieranie sesji użytkownika.1:05:28No tutaj też trudno czasami w literaturze1:05:31znaleźć dokładną informację jak to się odbywa.1:05:34Ja tu starałem się Państwu przygotować.1:05:39Teraz to zamkniemy, żeby nie przeszkadzało.1:05:43Bo to też jest ciekawe, jak to się realizuje.1:05:49Czyli jak wspomniałem, proces init generuje takie procesy gdty,1:05:56czyli używa funkcji systema fo, następnie xdek,1:06:00a celem procesu gdty jest próba otwarcia określonego terminala.1:06:07Czyli dla wszystkich tych terminali zadeklarowanych1:06:12w tym pliku init tab, o którym mówiłem przed chwilką,1:06:17generowane są kolejne procesy gdty1:06:21i ten proces gdty próbuje otworzyć terminal.1:06:27No jeśli jakiś użytkownik uruchomił terminal,1:06:32no to mu się to powiedzie i pójdzie dalej.1:06:35Jeśli terminal jest niewłączony przykładowo,1:06:39no to on usypia.1:06:42Informacja do tego trafia do init,1:06:46a init znowu go uruchamia i tak krótko.1:06:50Aż tak, żeby tylko użytkownik określonego terminala1:06:56będzie chciał używać, żeby miał do niego dostęp.1:06:59Teraz powiedzmy, że ta operacja się powiedzie,1:07:02że ten terminal jest w użyciu,1:07:06czyli ma włączone zasilanie i został uruchomiony.1:07:09Ale na co zwrócicie uwagę tutaj?1:07:13Proszę zobaczyć, że dla procesu gdty1:07:16wykonywana jest tylko funkcja systemowa exec.1:07:20Ale co to znaczy?1:07:21Exec login.1:07:23Że teraz, że to jest ten sam proces,1:07:26ma ten sam identyfikator,1:07:30a proces login to jest proces,1:07:33który umożliwia użytkownikowi zalogowanie do systemu.1:07:37Czyli wyświetla login, wygiela password,1:07:40prosi o wpis i sprawdza, czy to się zgadza, prawda?1:07:46Czy użytkownik ma prawo do zalogowania.1:07:49I jeśli...1:07:50Może się okazać błąd.1:07:52No jeśli błąd, to proszę zobaczyć,1:07:54ale jego procesem macierzystym, który jest, nie gdty.1:07:58Bo to jest ten sam proces, tylko wykonuje,1:08:00co ten, tylko wykonuje inny program.1:08:03Tylko jego macierzystym jest init.1:08:06I tak długo, aż...1:08:11No w końcu się...1:08:13Będzie moment, że dany użytkownik,1:08:16który tam będzie korzystał z tego terminala,1:08:19się prawidłowo zaloguje.1:08:21Dobrze.1:08:23No to wtedy mamy dalej1:08:26też tylko funkcja systemowa exec,1:08:29czyli ten sam proces teraz1:08:32z argumentem login shell,1:08:36czyli program login shell zastąpi program login,1:08:40ale w tym samym procesie.1:08:42To jest dalej ten sam proces,1:08:44którego to procesu procesem macierzystym jest init.1:08:49I ten login shell obsługuje sesję.1:08:54No i teraz, jeśli w trakcie sesji1:08:56użytkownik zdecyduje się o zakończeniu pracy,1:08:59no to będzie exit i powrót do init.1:09:03Ale jeśli teraz będzie wykonywał1:09:05pewne polecenie, zgłaszał pewne polecenie,1:09:08no to już mamy całą parę.1:09:11For exec.1:09:14Proces.1:09:15Wykonuje się proces wywołany poleceniem.1:09:17Ten usypia.1:09:19Jeśli on się skończy, to następuje1:09:22obudzenie tego shella.1:09:24Może kolejny program1:09:29uruchomić jako proces na żądanie użytkownika.1:09:33Czyli tak to widzimy działa.1:09:35I na co chcę zwrócić uwagę?1:09:38Właśnie, że mamy jeden proces,1:09:41który jest potomnym init,1:09:43który wykonuje kolejno.1:09:45Najpierw program getDTY,1:09:47potem program login, potem login shell.1:09:49I tu widzimy, dlaczego1:09:52mamy dwie funkcje systemowe.1:09:55Jedną, która służy utworzeniu nowego procesu,1:09:59a inną funkcję systemową, która służy temu,1:10:03żeby określony program1:10:05znalazł się w sekcji tekstu określonego procesu.1:10:10Tu widzimy istotne użycie tego właśnie,1:10:14tej możliwości.1:10:20Czyli tu widzimy, że tu mamy to właśnie,1:10:23że proces getty umożliwia otwarcie terminala.1:10:27On wywołuje proces login.1:10:29Czyli proces jest w zasadzie ten sam,1:10:31tylko inny program się dokonuje w ramach niego.1:10:33Ten sprawdza prawidłowe rozpoczęcie1:10:35i wywołuje, no właśnie,1:10:37tylko funkcja exec jest.1:10:39Czyli proces jest ten sam,1:10:41tylko uruchamia interpretator poleceń.1:10:45Ten uruchamia skrypt startowy.1:10:49Przykładowo ten,1:10:51to też, że to jest fiat pakarda,1:10:52to jest etc profile.1:10:54Tam mamy dwa skrypty startowe.1:10:56Jeden ogólny,1:10:58który jest w gestii administratora.1:11:02Może zmieniać ten ogólny profil1:11:04dla wszystkich użytkowników wspólnych.1:11:06Ale każdy użytkownik ma jeszcze swój skrypt home.profile1:11:14i to każdy użytkownik może niezależnie,1:11:17no to to jest domena danego użytkownika tutaj,1:11:22a może swój skrypt startowy modyfikować.1:11:26Przykładowo inaczej można sobie przyjąć postać1:11:29tej,1:11:31tego znaku zachęty generowanego przez login.1:11:34No i właśnie tu wyświetla znak zachęty,1:11:37ale ten znak zachęty może być inny.1:11:39Pewnie Państwo się tym bawili.1:11:41I on już interpretuje wprowadzane polecenia1:11:44i tworzy ewentualne procesy potomne,1:11:47jeśli trzeba.1:11:50No i jeszcze kilka słów o zamykaniu systemu.1:11:54Wspomniałem, że jest to bardzo istotne.1:11:57Czyli ta pierwsza rzecz,1:11:59która wymagana jest zakończenie wszystkich procesów.1:12:02I właśnie tu jest istotna ta synchronizacja,1:12:05o czym mówiłem,1:12:06i odmontowanie systemu pliku.1:12:11Czyli należy wywołać ten demon,1:12:16proces demon o nazwie sync,1:12:18o którym mówiłem,1:12:20który zapewni synchronizację informacji1:12:23między buforami,1:12:25a tym, co jest zapisane na dyska.1:12:28I wtedy należy odmontować system pliku1:12:31i wtedy ten program file system check1:12:34nie będzie musiał być uruchamiany, sprawdzany.1:12:37No, kolejno zatrzymanie procesora1:12:40i dopiero wtedy można wyłączyć zaśle.1:12:43Zwykle jest do tego odpowiedni1:12:46program administracyjny, który na to pozwala.1:12:49Tu podaję przykład takiego1:12:52programu polecenia shut down,1:12:56też wzięty z systemu,1:12:58na przykład placarda,1:13:01który realizuje te wszystkie cztery punkty1:13:06wymienione wyżej,1:13:08czyli całą operację zatrzymania systemu.1:13:11No przykładowo tutaj widzimy1:13:14to 300 oznacza czas,1:13:17który od tego momentu upłynie,1:13:19po którym zostaje wyłączenie systemu.1:13:22To zostaje przekazane do wszystkich użytkowników,1:13:24ta informacja.1:13:26Czyli użytkownicy mają 5 minut1:13:28przykładowo w tym momencie na to,1:13:30żeby pozamykać swoją pracę,1:13:33pozapisywać wszystkie informacje.1:13:38Dobrze, na tym chcę zakończyć informację1:13:41dotyczącą zarządzania procesami,1:13:45gdzie jak wspomniałem,1:13:47istotne są informacje, w szczególności1:13:49samorozumienie procesów,1:13:51jak są tworzone, wykonywane,1:13:54jak obsługiwana jest sesja,1:13:56przez Login Shell,1:13:59jak są nowe procesy tworzone w trakcie pracy.1:14:03No i istotne jest to zagadnienie1:14:05szeregowania procesów,1:14:07jak to obecnie wygląda na tle1:14:11wcześniejszych różnych idei szeregowania.1:14:17Przejdziemy teraz do następnego zagadnienia1:14:20nowego działu.1:14:26Dotyczącego zarządzania,1:14:35tym razem pamięcią operacyjną.1:14:39Ze względu na ograniczony czas,1:14:41zachęcam Państwa oczywiście1:14:43za każdym razem do korzystania z literatury.1:14:45Tu przykładowo książce Zyli Belszaca,1:14:48Petersona Garwina rozdział 7-8,1:14:50w książce Bacha rozdział 9.1:14:53Natomiast popatrzmy na to,1:14:55popatrzmy na parę tutaj informacji podstawowych.1:15:01Mamy wiele procesów działających w systemie.1:15:05Żeby proces się mógł wykonywać,1:15:08no to rozkazy, które mają być wykonywane1:15:11przez procesor1:15:13i wszystkie argumenty tych rozkazów1:15:16muszą się znajdować w pamięci operacyjnej.1:15:20Problem,1:15:22która nosi nazwę czasami też1:15:24w pamięci fizycznej,1:15:26no nie czasami,1:15:27tylko wymiennie można powiedzieć.1:15:29Można ją interpretować w ten sposób,1:15:31że jest taka tablica bajtów,1:15:33w której każdy element ma swój jednoznaczny adres.1:15:38Mówimy o adresie fizycznym w pamięci operacyjnej.1:15:44Znaczy jeszcze raz,1:15:45proces żeby mógł się wykonywać,1:15:47on musi się znajdować przynajmniej częściowo1:15:49w pamięci operacyjnej.1:15:51My wiemy już,1:15:52że1:15:54w pamięci operacyjnej cały czas1:15:56jest załadowany system operacyjny.1:15:58No tak się proces ma.1:15:59Czyli1:16:00on już zajmuje pewną część pamięci operacyjnej,1:16:03ta pozostała część jest tylko dostępna1:16:05dla procesu.1:16:07No tutaj1:16:08się mieści no i samo1:16:10jądro, ale wszystkie struktury systemu operacyjnego1:16:12to tam są właśnie przechowywane1:16:14w tej części pamięci.1:16:19Dla prawidłowego rozumienia współpracy,1:16:22no,1:16:23różnych1:16:25i działania różnych algorytmów1:16:27związanych z zarządzaniem pamięcią,1:16:29chcę tutaj jeszcze przypomnieć Państwu,1:16:33jak1:16:34procesor współpracuje z pamięcią operacyjną.1:16:48No zakładam, że Państwo jeszcze pamiętają1:16:50z pierwszego roku pewne informacje,1:16:52tak dla,1:16:53dla pewności.1:16:54Czyli mamy tak,1:16:55mamy procesor połączony szyną1:16:57z pamięcią operacyjną.1:17:00Szyna od strony fizycznej,1:17:02jest to pewna linia,1:17:03pewna liczba linii1:17:05łączących procesor z pamięcią,1:17:09ale jest także pewien protokół1:17:11obsługi tej szyny.1:17:13To jest kolejna rzecz.1:17:16I teraz tak,1:17:18wyróżniamy trzy grupy linii1:17:20w ogólnym wypadku.1:17:21Linie adresowe,1:17:23linie danych,1:17:24linie sterujące.1:17:25Jak wspomniałem,1:17:26każdy element pamięci operacyjnej1:17:28ma swój unikalny adres.1:17:32Jak wygląda teraz współpraca1:17:34procesora z pamięcią operacyjną?1:17:36No procesor, jeśli chce1:17:40za,1:17:41pobrać,1:17:42no powiedzmy pobrać,1:17:43odczytać1:17:45pewien element pamięci operacyjnej.1:17:47Musi w liniach sterujących podać odczyt,1:17:50w liniach adresowych podać adres,1:17:54spod którego pamięć ma dokonać odczytu1:17:58tego, co się w danym,1:18:00w danym elemencie znajduje.1:18:02A pamięć operacyjna w liniach danych1:18:05podaje wynik odczytu,1:18:08który odczytuje teraz procesor.1:18:10W ten sposób dostaje informację1:18:12z pamięci operacyjnej.1:18:14No przykładowo,1:18:15gdyby chciał wykonać określony rozkaz,1:18:18który ma powiedzmy1:18:19trzy argumenty,1:18:23chcemy dodatkowanie zrobić,1:18:25no to trzeba pobrać rozkaz,1:18:27pierwszy składnik, drugi składnik,1:18:29czyli trzy elementy trzeba pobrać1:18:31z pamięci operacyjnej,1:18:33prawda, w ten sposób.1:18:35A teraz wynik,1:18:36chce zapisać z pamięci operacyjnej,1:18:38to jak teraz zapis będzie wyglądał?1:18:40W liniach adresowych adres,1:18:44w liniach sterujących zapis,1:18:46informacja, że to jest zapis,1:18:47a w liniach danych to,1:18:49co pamięć operacyjna zapisze.1:18:53No pamięć operacyjna,1:18:54jedyne działanie,1:18:55które może wykonać,1:18:56to odczytać pod określonego adresu coś,1:19:00lub pod określony zapis,1:19:03lub pod określony adres coś zapisać.1:19:06Pamiętajmy o tym.1:19:08To jest istotne dla rozumienia1:19:10dalszych działań,1:19:11które będę tutaj omawiał.1:19:15Jeśli teraz popatrzymy na miejsce1:19:18w pamięci operacyjnej,1:19:20które to zajęte,1:19:22to jest to przez jądro,1:19:24pewne miejsce jest wolne1:19:26na procesy użytkowników,1:19:28to w ogólnym wypadku1:19:30tego miejsca dla procesów1:19:32użytkowników jest za mało.1:19:34To znaczy,1:19:36pierwsza rzecz,1:19:38ogólnie1:19:40nie wszystkie procesy1:19:42znajdą tam miejsce,1:19:43nie ma możliwości umieszczenia,1:19:45bo mówimy o komercyjnych systemach,1:19:47to jest jedna rzecz.1:19:49A druga rzecz jest jeszcze taka,1:19:51że mogą się zdarzyć procesy,1:19:53które wymagają więcej1:19:55tej pamięci operacyjnej,1:19:57niż ona w ogóle istnieje.1:20:01No i powstaje problem,1:20:03jak to rozwiązać.1:20:05I tutaj wymieniłem1:20:07takie trzy idee.1:20:09Metoda nakładek,1:20:11metoda użytkowników,1:20:13metoda użytkowników,1:20:15metoda nakładek,1:20:17wymiana procesu,1:20:19strońcowanie.1:20:21To od razu powiem,1:20:23że w dalszym ciągu1:20:25najwięcej będziemy mówili1:20:27o tym, co się obecnie stosuje,1:20:29to są idee strońcowania,1:20:31ale też o tych1:20:33też trzeba wspomnieć,1:20:35bo stanowią pewne wprowadzenie1:20:37do tej idei.1:20:39Na czym polega nakładek?1:20:41Też przygotowałem Państwu1:20:43obrazek,1:20:45obrazek1:20:51pokazujący,1:20:53jak można1:20:57wykonać1:20:59program1:21:01jako proces,1:21:03który się nie mieści1:21:05w pamięci operacyjnej.1:21:07Tu jako przykład,1:21:09no właśnie tej idei nakładkowania1:21:11mamy program assemblera,1:21:13w którym można wyróżnić1:21:15dwa kody1:21:17pierwszego kodu1:21:19i drugiego przebiegu,1:21:21w którym to1:21:23pierwszy kod przebiegu1:21:25tworzy pewną tablicę symboli,1:21:27no to trzeba ją zapisać1:21:29w pewnym miejscu1:21:31w pamięci operacyjnej,1:21:33natomiast na podstawie1:21:35tej tablicy symboli1:21:37kod drugiego przebiegu1:21:39generuje wynik.1:21:41Popatrzmy na to,1:21:43gdzie jest miejsce1:21:45w okolicach.1:21:47Na kod pierwszego przebiegu,1:21:49na kod drugiego przebiegu,1:21:51na tą tablicę symboli,1:21:53no są pewne wspólne programy1:21:55realizujące oba te działania.1:21:57Czyli razem potrzeba1:21:59200 kilobajtów.1:22:01A wyobraźmy sobie,1:22:03że mamy teraz1:22:05pamięć operacyjną,1:22:07której wolne miejsce1:22:09wynosi tylko 150 kilobajtów.1:22:11Co można zrobić?1:22:13Można napisać program1:22:15nakładkowania1:22:17i tutaj widzimy1:22:19tablica symboli będzie1:22:21w pamięci operacyjnej,1:22:23wspólne programy 30 kilobajtów,1:22:25program obsługi nakładek,1:22:27zostaje nam 901:22:29i tak, najpierw wpiszemy kod1:22:31pierwszego przebiegu, który zajmie 70,1:22:33mieści się,1:22:35wykona się,1:22:37wyrzucamy go,1:22:39a w jego miejsce,1:22:41czyli można powiedzieć1:22:43w jego miejsce,1:22:45nałożymy kod drugiego przebiegu,1:22:47który wymaga 80 kilobajtów1:22:49i też się może wykonać.1:22:51Czyli mamy dwie nakładki,1:22:53tablica symboli,1:22:55wspólne programy1:22:57i kod pierwszego przebiegu,1:22:59no plus ten program obsługi nakładek,1:23:01który musi działać cały czas,1:23:03a teraz nakładka na to,1:23:05tablica symboli,1:23:07wspólne programy,1:23:09kod drugiego przebiegu,1:23:11czyli to jest idea,1:23:13która pokazuje,1:23:15że można wykonać proces,1:23:17nawet gdy nie wszystkie1:23:19informacje1:23:21do jego wykonania są potrzebne,1:23:23jeśli jest możliwość taka,1:23:25że1:23:27pewną część tego programu1:23:29wyrzucimy z pamięcią1:23:31operacyjnej, a1:23:33kolejną w to miejsce,1:23:35nie wiem czy zrobimy taką właśnie nakładkę,1:23:37no ale istotne jest to,1:23:39pamiętajmy, że1:23:41żeby1:23:43proces się mógł wykonać,1:23:45żeby procesor mógł wykonać1:23:47określony rozkaz, to wszystkie1:23:49informacje związane z tym rozkazem1:23:51muszą być w pamięci operacyjnej.1:23:55No dobrze, widzę, że niestety1:23:57ten czas na dzisiejszy1:23:59się skończył,1:24:01także na tej metodzie1:24:03nakładek zakończę.1:24:05Ona jest taką ilustracją1:24:07do lepszego rozumienia1:24:09tej1:24:11idei, obecnie stosowanej1:24:13idei strojnicowania procesu.1:24:15Dobrze, także1:24:17dziękuję Państwu za uwagę.1:24:19Ja tutaj1:24:21zakończę nagrywanie.