

Etap	1	2	3	4	5	Suma
Punkty	3	7	4	6	4	24
Wynik						

L2: Hauptadministrationsbüroangestelltenhierarchie

W roku 1910 administracja austro-węgierska w Pradze rozkwita, osiągając apogeum swojej doskonałości. Gmach Głównego Urzędu Cesarsko-Królewskiej Administracji wznosi się dumnie nad miastem. Dzięki nowoczesnym technologiom, takim jak telegraf, telefony i maszyny do pisania, praca urzędników staje się coraz bardziej efektywna. Zredukowano czas niezbędny na skreślanie wniosków i nanoszenie pieczęci „ABGELEHNT” do zaledwie kilku dni roboczych.

Nie wszystko jednak idzie zgodnie z planem. W wyniku niepokojącego błędu systemowego odsetek wniosków rozpatrywanych pozytywnie wzrósł do alarmujących 2,3%. Ministerstwo Spraw Wewnętrznych w Wiedniu uznało to za skandal i zagrożenie dla porządku biurokratycznego Monarchii. W celu uszczelnienia procedur i skuteczniejszego eliminowania niepożądanych wniosków podjęto decyzję o wdrożeniu nowej struktury administracyjnej.

Nowy model organizacyjny zakłada, że każdy dokument musi przejść przez wielostopniowy system weryfikacji. W praktyce oznacza to, że zanim petent otrzyma brakującą stronę paszportu, jego sprawa musi zostać przekazana przez niezliczone biurka, przekartkowana, ostemplowana, a w końcu—odłożona na półkę w archiwum, gdzie zniknie na zawsze.

Aby zapewnić skuteczność nowej procedury, Wydział Modernizacji Administracji postanowił wdrożyć system testowy w oparciu o najnowsze osiągnięcia w dziedzinie maszyn liczących. Cała symulacja zostanie przeprowadzona przy użyciu wieloprotokolowej architektury obliczeniowej, a przepływ dokumentów będzie odwzorowywany poprzez system sygnałów przesyłanych między procesami reprezentującymi poszczególnych urzędników.

Celem testu jest stworzenie idealnej maszyny biurokratycznej, w której żaden wniosek nigdy nie zostanie rozpatrzony pozytywnie, a każda decyzja pozostanie w zawieszeniu na bliżej nieokreślony czas.

Pamiętaj o sprawdzaniu błędów zgłaszanych przez wywoływane funkcje oraz o zwalnianiu zaalokowanych zasobów takich jak pamięć i deskryptory.

Etapy:

- 3 p.** Proces po uruchomieniu weryfikuje, że podane zostały mu dwa argumenty. Pierwszym z nich będzie ścieżka do katalogu z plikami, które będą reprezentować urzędników. Drugim będzie nazwa najwyższego urzędnika. W tym etapie nie są one jeszcze używane. Proces uruchamiaj poleceniem `./sop-admin tree Franz_Joseph`.
Proces ustawia maskę sygnałów blokującą SIGUSR1, SIGUSR2 i SIGINT, wypisuje `Waiting for SIGUSR1` i czeka na dostarczenie SIGUSR1. Po jego otrzymaniu kończy się.
- 7 p.** Całość tego etapu zawrzyj w jednej funkcji. Będzie ona wywoływana rekurencyjnie w kolejnych etapach. Musi ona przyjmować przynajmniej jeden argument: `char* name`, który jest nazwą pliku w katalogu podanym jako pierwszy argument programu. Funkcja ta najpierw wypisuje `My name is <name> and my PID is <PID>`, otwiera plik `<argv[1]>/<name>` i odczytuje jego zawartość używając niskopoziomowego API. Nazwa pliku to imię urzędnika, który jest reprezentowany przez ten plik. Plik zawsze zawiera dwie linijki tekstu. W obu liniach znajduje się imię podwładnego tego urzędnika, lub znak -, reprezentujący brak podwładnego. Dla każdego podwładnego zdefiniowanego w pliku funkcja wypisuje `<nazwa pliku> inspecting <imię podwładnego>`.
Potem wypisuje `<nazwa pliku> has inspected all subordinates` i się kończy. Wywołaj napisaną funkcję w funkcji `main`.
- 4 p.** Każdorazowo po wypisaniu `<nazwa pliku> inspecting <imię podwładnego>` w poprzednim etapie, wywołaj `fork()`. Proces rodzic w tym momencie kontynuuje pracę z kolejnymi podwładnymi, a proces dziecko wywołuje implementowaną w poprzednim etapie funkcję, przekazując

jej imię podwładnego jako argument. Oczekiwanym wynikiem działania programu na tym etapie jest utworzenie hierachii procesów, odzwierciedlającej strukturę organizacyjną administracji. Na kartce z poleceniem znajdziesz graficzną reprezentację struktury ustalonej w katalogu `tree`.

Po przetworzeniu wszystkich podwładnych, proces czeka na ich zakończenie, wypisuje
`<name> is leaving the office`
 i kończy się.

4. 6 p. Zaimplementuj logikę przetwarzania dokumentów. Zamiast kończyć się po przetworzeniu wszystkich podwładnych, proces zaczyna oczekiwać na sygnał `SIGUSR2`. Po otrzymaniu go, losowo podejmuje jedną z dwóch akcji. Z prawdopodobieństwem $\frac{1}{3}$ wypisuje
`<name> received a document. Sending to the archive,`
 i kontynuuje czekanie na sygnały. Z prawdopodobieństwem $\frac{2}{3}$ wypisuje
`<name> received a document. Passing on to the superintendent,`
 wysyła `SIGUSR2` do procesu rodzica i kontynuuje czekanie na sygnały. W drugim z tych przypadków, jeśli proces jest najwyższym urzędnikiem, wypisuje tylko `<name> received a document. Ignoring.`
5. 4 p. Zaimplementuj logikę opuszczania biura. Kiedy którykolwiek proces otrzyma sygnał `SIGINT`, to wypisuje
`<name> ending the day.,`
 wysyła `SIGINT` do swoich dzieci, czeka na ich zakończenie, wypisuje
`<name> is leaving the office`
 i kończy się.

