

Sprawozdanie SK2

Temat: Zdalne zamykanie systemów operacyjnych.

1. Opis projektu

Projekt polegał na stworzeniu **systemu zdalnego zarządzania węzłami sieciowymi**, w którym połączony z centralnym serwerem **klient** może **monitorować i wyłączać agentów** działających na różnych komputerach. Każdy węzeł uruchamia program-agenta, który łączy się z serwerem, dostaje pierwszy wolny ID, a następnie zaczyna okresowe przesyłanie **heartbeat'ów** jednocześnie nasłuchując ewentualnego polecenia ze strony serwera (np. wyłączenie systemu). Do serwera podłącza się również klient który może pobrać listę wszystkich agentów wraz z ich aktualnym stanem (online/offline) i zdalnie wyłączyć wskazany węzeł.

W prezentowanym rozwiążaniu wykorzystany został **język C** oraz **protokół TCP/IP** do komunikacji sieciowej. Implementacja bazuje na socketach, funkcjach **select()** do obsługi wielu połączeń jednocześnie i prostym systemie identyfikacji połączeń za pomocą enumów i tablic.

2. Opis komunikacji

Komunikacja opiera się na **połączeniu TCP**. Każdy agent lub klient połączywszy się z serwerem wysyła pierwszą wiadomość określającą **typ połączenia** (AGENT lub CLIENT). **Serwer przydziela agentom unikalne identyfikatory** i zapisuje ich deskryptory w tablicy.

Serwer wykorzystuje mechanizm **select()**, który pozwala na jednoczesne obsłużenie wielu połączeń bez blokowania programu. Dzięki temu może reagować zarówno na heartbeat'y od agentów, jak i na przychodzące żądania klienta.

Agenci przesyłają **regularnie heartbeat**, który pozwala serwerowi monitorować ich aktualny stan. Jeśli przez określony czas serwer nie otrzyma heartbeat'u od danego agenta, zostaje on uznany za wyłączony (**timeout**).

Klient może wysłać komendę LIST, aby pobrać **stan wszystkich agentów** lub KILL <ID>, aby zdalnie **wyłączyć wskazanego agenta**. Serwer interpretuje te żądania i odpowiednio reaguje, np. przesyła listę agentów lub wysyła komendę SHUTDOWN do wybranego agenta.

3. Podsumowanie

Projekt stanowi praktyczną realizację **systemu zdalnego zarządzania węzłami sieciowymi** z wykorzystaniem **języka C** oraz **protołu TCP/IP**. Głównym celem było stworzenie architektury, w której klient, poprzez centralny serwer, może monitorować i kontrolować wielu agentów, działających na różnych komputerach.

Kluczowym elementem jest **mechanizm komunikacji oparty na socketach i funkcji select()**, który umożliwia serwerowi jednoczesną obsługę wielu połączeń bez blokowania działania programu. Dzięki temu serwer może w tym samym czasie reagować zarówno na heartbeat'y od agentów, jak i na komendy przychodzące od klienta. Funkcja **select()** jest również istotna z perspektywy agenta, który właśnie dzięki niej jest w stanie równolegle raportować swój stan do serwera i stale nasłuchiwać czekając na potencjalne polecenie.

Dzięki zastosowaniu m.in. TCP/IP, socketów oraz mechanizmu **select()** projekt umożliwia **sprawne i niezawodne zarządzanie** podłączonymi węzłami sieciowymi, zapewniając jednocześnie **pełną kontrolę** nad ich stanem w **czasie rzeczywistym**.