

Projekt X

Adam Kraczkowski

Państwowa Wyższa Szkoła Zawodowa w Chełmie

Streszczenie

Poniższa publikacja ma za zadanie przedstawienie sposobu budowy graficznych stacji roboczych.

1. WSTĘP

Od kilku lat można zauważyć bardzo szybki rozwój informatyki i jej ingerencję w codzienne życie. Rozwój ten powoduje, że coraz to nowsze aplikacje wymagają coraz to większej mocy obliczeniowej i tym samym zmuszają nas do ciągłej rozbudowy sprzętu. W tym pędzie za nowościami zapominamy często o zdrowym rozsądku i rozwiązaniach, które istnieją od lat, ale umknęły nam i nawet nie wiemy o ich istnieniu. Dzieje się tak głównie z powodu licznych kampanii reklamowych, które zamazują wizerunek komputera, jakiego na prawdę potrzebujemy, tym samym nakręcając koniunkturę. Rozwiązania, które są dość stare, ale przez to sprawdzone i wydajne, rozwiązania potrafiące pogodzić nasz pęd z nowymi programami i jednocześnie dość znacznie ograniczyć konieczność rozbudowy komputerów na naszych biurkach.

Rozwiązania okazują się istnieć od dawna jako tzw. usług terminalowe działające na platformie systemu operacyjnego Unix i środowiska X Window System.

Informacje teoretyczne

Większość komputerów używanych do codziennej pracy to jednostki o dość dobrych parametrach. Jednostka centralna to na ogół nowy, lub zaledwie 2-3 letni komputer, z dużą ilością pamięci operacyjnej i co najmniej kilku gigabajtowym dyskiem twardym. Taka konfiguracja jest niezbędną do komfortowej i zarazem wydajnej pracy. Jednak jeśli przyszło by nam wyliczyć średnie wykorzystanie procesora w tym komputerze to mogło by się okazać, że nie przekracza ono 5%. Podobnie jest z przestrzenią dyskową. Nagminnie na dyskach zalega wiele nieużywanych kopii dokumentów czy plików multimedialnych.

Kolejnym "grzechem" jest nie przestrzeganie zasad tworzenia kopii zapasowych. Na ogół nie dysponujemy gotowymi automatami, głównie z uwagi na ich cenę, a "ręczne" kopiowanie ważnych plików jest uciążliwe i na ogół nie wykonywane systematycznie przez personel.

Sporo kłopotu sprawia też nieustanna konieczność aktualizacji składników systemu i programów użytkowych. Niejednokrotnie administrator jest zmuszony do "odwiedzenia" kolejno wszystkich stanowisk pracy i zainstalowanie tam niezbędnych poprawek i aktualizacji. Co jest czasochłonne.

Wielokrotna instalacja tych samych aplikacji jest też czystym marnotrawstwem miejsca na dyskach twardych. Jeżeli dany program można wielokrotnie uruchomić to czy nie można go też raz zainstalować?

Idea budowy terminala

Większość z powyższych "utrapien" administratora znika lub staje się mniej uciążliwymi, jeśli pracownicy zamiast "pełnych" komputerów mają na swoim biurku terminale.

Terminal jest rozwinięciem idei urządzeń wejścia/wyjścia, jest to komputer spełniający rolę końcówki sieciowej. Osoba pracująca przy terminalu korzysta z zasobów serwera. Wszelkie programy są uruchamiane na serwerze. Uruchomione programy korzystają z procesora, pamięci operacyjnej, dysków serwera. Terminal służy jedynie do komunikacji użytkownika z serwerem.

Dzięki temu uzyskujemy scentralizowane zarządzanie każdą końcówką. Aktualizacja oprogramowania odbywa się centralnie w jednym miejscu i program jest instalowany tylko raz, tym samym oszczędzamy i miejsce na dysku i czas potrzebny na zainstalowanie nowej wersji jakiegoś programu, poprawieniu jego konfiguracji czy wsunięciu błędów w działaniu. Tworzenie kopii zapasowych jest proste i sprowadza się do skopiowania jednego katalogu, w którym są katalogi domowe użytkowników. Administracja kontami, nadawanie i odejmowanie uprawnień odbywa się w jednolity, spójny sposób. Administrator ma pełny wgląd (oczywiście, jeśli tego potrzebuje), co użytkownik ma na dysku, ile miejsca używa i z jakich programów korzysta. Jeśli pracownik zostanie przeniesiony do innego działu lub zwolniony, administrator może natychmiast zabronić użytkownikowi logować się do serwera i tym samym niweluje groźbę wyrządzenia przez niego szkód firmie czy systemowi. Użytkownicy natomiast nie są na sztywno przywiązani do jednego biurka i komputera, na którym pracują, w każdej chwili mogą wylogować się z zajmowanego stanowiska pracy, przejść do drugiego (np.: na innym piętrze), aby tam po zalogowaniu móc kontynuować swoją pracę na swoich plikach.

Przy okazji mamy tu do czynienia ze swego rodzaju paradoksem. Okazuje się bowiem, że nasz serwer jest w stanie obsłużyć jednocześnie więcej klientów niż mogło by się to z początku wydawać. Jeśli natomiast do serwera jest aktualnie zalogo-

wanych mniej użytkowników niż zakładaliśmy przy projektowaniu serwera to uzyskują oni większą moc obliczeniową. Tym samym serwer jest rzadziej bezczynny i lepiej wykorzystywany.

Dodatkowo, jeśli nasze potrzeby, co do mocy obliczeniowej serwera z upływem czasu wzrosną to rozbudowujemy jedynie serwer.

Historia terminali

W początkowej fazie rozwoju informatyki jedynymi dostępnymi terminalami, używanymi zresztą do dziś, były terminale tekstowe. Dana instytucja wydawała ogromne pieniądze na kupno centralnego komputera, który niejednokrotnie miał wydzielone własne pomieszczenie z klimatyzacją i specjalnym zasilaniem, tylko po to, by go tam zamknąć i pilnować aby nikt niepowołany się tam nie dostał.

Terminale łączyły się na ogół z serwerem za pomocą specjalnej magistrali. Wraz ze stopniowym rozwojem sieci LAN i WAN do komunikacji zaczęto stosować protokół TCP/IP. Okazało się tym samym, że można za pomocą sieci internet pracować z miejsca oddalonego o tysiące kilometrów.

Pracownicy, aby skorzystać z głównego komputera mieli do dyspozycji terminale tekstowe podłączone do serwera. Za ich pomocą wykonywali swoją codzienną pracę, nie mając nawet pojęcia co faktycznie działa na drugim końcu kabla łączącego terminal z serwerem, ta wiedza nie była im zresztą potrzebna.

Dzięki prostocie budowy terminala były to urządzenia trwałe, ciche i tanie w eksploatacji. Ich konstrukcja nie wymagała specjalnych pomieszczeń i układów chłodzenia.

Główną wadą tego rozwiązania była groźba awarii centralnego komputera lub naruszenie zasad bezpieczeństwa firmy przez niepowołany dostęp do poufnych danych. Aby poradzić sobie z tymi zagrożeniami wprowadzono systemy logowania i nadawania uprawnień zalogowanemu użytkownikowi, zadbano też o to aby system operacyjny serwera był stabilny i niezawodny.

Coraz większa popularność systemów "okienkowych", czyli z graficznym interfejsem użytkownika jak MS Windows czy Mac OS spowodowała zapotrzebowanie na terminale mogące pracować również w trybie graficznym.

X Window System

Zbiór protokołów, jakim jest X Window System jest otwartym standardem niezależnym od systemu operacyjnego i sprzętu. Pozwala na zbudowanie w oparciu o niego rozproszonego systemu okienkowego. X Window System jest oparty na technologii klient-serwer. X Server to program, który działa na maszynie użytkownika. Akceptuje on połączenia z różnych maszyn, wyświetlając okienka. Po sieci przekazywane są informacje o kliknięciu myszką, zdarzeniach klawiatury - od X Servera do X Clienta. X Client jest to program napisany dla X. On z kolei wysyła do X Servera polecenia narysowania okienka, itd.

X Consortium (grupa dawniej zarządzająca rozwojem X) żądała by w stosunku do tego oprogramowania używać następujących określeń:

X

X Window System

X Version 11

X Window System, Version 11

X11

Oryginalny kod systemu X Window został stworzony w MIT (ang. *Massachusetts Institute of Technology*), później komercyjne firmy zaakceptowały go jako przemysłowy standard na platformę Unix. XFree86 jest darmowym, do dowolnego rozpowszechniania przeniesieniem MIT X Window System X11R6 na systemy 80386/80486/ Pentium, pierwotnie zaprojektowanym przez zespół programistów pod przewodnictwem Davida Wexelblata.

Implementacja ta znana jako XFree86 jest dostępna na SystemV/386, 386BSD, oraz innych Unix'ów x86, w tym Linux'a. Dystrybucja zawiera wszystkie potrzebne narzędzia, biblioteki i pliki.

Podstawową różnicą w stosunku do innych systemów GUI (ang: Graphical User Interface) jest przezroczystość dla sieci. Sam serwer X11 nie zapewnia nic poza obszarem do wyświetlania, wywołaniami systemowymi oraz (częściowo) metodami autentyfikacji (tak sesji użytkownika na zdalnych maszynach, jak i sesji klienckich na lokalnym terminalu). Za wygląd środowiska pracy, sterowanie zdarzeniami odpowiada specyficzna klasa klientów zwana zarządcami okien (window manager). Innym elementem składającym się na działanie X są klasy tzw. widgetów czyli bibliotek two-

rzących typowe elementy wyświetlane przez wm. Jednym z bardziej znanych zestawów window manager i klasy widgetów jest Motif wytwarzany przez wielu producentów --- jest też darmowa implementacja o nazwie Lesstif. W oparciu o Motifa stworzono środowisko pracy CDE. Innymi znanymi window managerami są: NeXTStep, fvwm czy modne ostatnio środowiska (a więc nie tylko wm ale i inne elementy) K Desktop Environment. i Gnome.

W podręczniku systemu UNIX czytamy "X Window System jest 'przezroczystym sieciowo' systemem okienkowym, który działa na dużej liczbie komputerowych i graficznych maszyn. Powinno być stosunkowo prosto skompilować oprogramowanie z X Consortium na większości systemów zgodnych z C i POSIX. Dostępne są również komercyjne implementacje na dużej liczbie platform.

2. SPOSÓB REALIZACJI

Konstrukcja serwera

Wbrew pozorom serwer nie musi być dużą maszyną. Jego konfiguracja zależy oczywiście od pracy, jaką mamy zamiar wykonywać. Należy jednak pamiętać, iż uruchomienie jakiegoś programu przez jednego użytkownika powoduje, że drugi użytkownik uruchomi ten program już o wiele szybciej. Dzieje się tak, dlatego, że dzisiejsze aplikacje potrafią współdzielić swoje biblioteki. Do pamięci nie jest ładowany cały drugi program lecz tylko jego część, kilkukrotnie mniejsza od całości. Podobny efekt każdy z nas zapewne zauważył w momencie włączania nowego dokumentu np.: OpenOffice, przy pierwszym uruchomieniu trwa to długo, ale przy otwieraniu równoległe drugiego dokumentu jest już bardzo szybkie. Analogicznie dzieje się w przypadku uruchamiania danej aplikacji na serwerze. Tylko, że drugi użytkownik uruchamiający w tym samym czasie ten sam program, będzie miał wrażenie, że serwer ma o wiele większą moc obliczeniową niż ma to naprawdę miejsce.

Wymagania, co do pojemności dysku stają się też mniejsze. Pojemność dysku może być taka sama jak w zwykłej stacji roboczej. Oszczędność miejsca bierze się stąd, że nie instalujemy już wielokrotnie tego samego programu. Zalecane jest jednak, aby był to dysk szybki, zdolny na obsłużenie wiele jednoczesnych żądań zapisu/odczytu. Powinno to być też urządzenie niezawodne.

Przy tego typu rozwiązaniach wzrasta natomiast zapotrzebowanie na dostępną pamięć operacyjną. Każda kolejna instancja tego samego programu potrzebuje pewną część pamięci. Im więcej pracujących programów, tym zapotrzebowanie na pamięć operacyjną większe. Jej wyczerpywanie spowoduje nadmierne wykorzystywanie pliku wymiany na dysku i duży spadek wydajności. W ekstremalnych warunkach może doprowadzić do niemożności pracy niektórych programów. Jednym słowem, aby praca mogła być prowadzona niezależnie (bez widocznych spadków wydajności) przez wiele stanowisk podpiętych do serwera, serwer musi mieć dość dużo pamięci operacyjnej. Ilość pamięci jest nawet ważniejsza od szybkości procesora.

Urządzeniem, którego na ogół nie ma w standardowym komputerze, jest napęd taśmowy. Jego zakup wiąże się z pewnym wydatkiem, ale może być bardzo pomocny w tworzeniu automatycznych kopii zapasowych.

Konstrukcja terminala

Aby stać się posiadaczem terminala graficznego możemy zrobić dwie rzeczy. Po pierwsze możemy kupić gotowe urządzenie, które po wyjęciu z pudełka można będzie od razu podłączyć do sieci i zacząć pracę. Producentami gotowych terminali graficznych są również polskie firmy. Drugi sposób jest trochę bardziej skomplikowany. Polega on na samodzielnym zbudowaniu terminala w oparciu o komputer klasy PC.

Terminal jest urządzeniem, którego zadaniem jest jedynie wyświetlanie „okienek”, więc nie ma wielkich wymagań, co do jego mocy obliczeniowej. Z powodzeniem można zastosować tu stary i już nieużywany komputer. W tym celu potrzebujemy jedynie: komputera z 16 MB RAM, procesorem klasy 80486 DX2/ 66MHz, wyposażonym dodatkowo jedynie w kartę sieciową, która umożliwi start z sieci.

Zalecany jest jednak zakup dobrego monitora. Podyktowane jest to z uwagi na słabe parametry techniczne starych kineskopów i tym samym szybkie męczenie wzroku przy pracy.

Nasz terminal nie potrzebuje ani dysku twardego ani stacji dyskietek. Jeśli dodatkowo nie wymaga wiatraka na procesorze to jest to urządzenie bardzo ciche. Co jednak najważniejsze jest też bardzo tanie.

Jak rozpocząć pracę?

Cały proces konfiguracji naszego komputera do pracy jako terminal graficzny jest opisany na stronie WWW projektu **Linux Terminal Serwer Projekt**, znajduje się tam szczegółowy opis konfiguracji terminali i serwera, oraz niezbędne oprogramowanie. Projekt dostępny jest pod adresem www.ltsp.org.

Pewne kłopoty może sprawić nam konfiguracja karty sieciowej, tak, aby mogła służyć do startu komputera z sieci. Musi być to karta wyposażona w specjalne gniazdo służące do montażu pamięci FLASH. Pamięć taką można kupić w sklepie elektronicznym za kilka złotych. Problemem może się jednak okazać zapisanie do tej pamięci

oprogramowania sterującego kartą. Wymaga to specjalnego programatora i pewnej znajomości nt. elektroniki. Jeśli nie mamy ochoty programować karty, będziemy zmuszeni umieścić w terminalu stację dyskiety i z niej inicjować start komputera.

Jak to działa w praktyce?

Po włączeniu zasilania BIOS komputera zaczyna szukać urządzenia w systemie, które przejmie dalej proces uruchamiania komputera. W naszym przypadku będzie to karta sieciowa. Rozsyła ona zapytanie protokołu DHCP o numer IP i położenie jądra systemu operacyjnego. Serwer przechwytuje zapytanie i w odpowiedzi podaje potrzebne dane. Za pomocą usługi o nazwie TinyFTP (*uproszczony protokół FTP, niewymagający autoryzacji*) przesyłane jest jądro systemu do terminala, po zakończeniu tej operacji rozpoczyna się właściwy proces startu systemu operacyjnego.

System przy starcie podmontowuje, wyeksportowany na serwerze przez NFS (*Network File System*) główny system plików i uruchamia X serwer. Oczom użytkownika ukazuje się zapytanie o login i hasło. Cały proces startu komputera trwa kilka sekund.

3. WNIOSKI

Stosowanie terminali w codziennej pracy może pozwolić na znaczne obniżenie kosztów wyposażenia stanowisk pracy. Praca administratora staje się bardziej wydajna.

Kłopotliwe może się okazać uniemożliwienie podsłuchu przez osoby trzecie przesyłanych informacji, można w tym celu wykorzystać protokół ssh do szyfrowania przesyłanych danych.

Wyzwaniem może się też okazać instalacja np.: napędu płyt kompaktowych czy konfiguracja karty dźwiękowej działającej na terminalu, w sposób przezroczysty dla użytkownika. Jednak twórcy projektu LTSP przygotowali na tą okoliczność szereg dodatkowych programów. Podobnie jest z konfiguracją drukarek.

Ponieważ X Serwer może wyświetlać jedynie okienka przesyłane z aplikacji przez sieć, niemożliwym staje się używanie aplikacji, które potrzebują dużej przepływności strumienia video. Nie możemy uruchamiać na terminalu rozbudowanych gier czy oglądać dynamicznych filmów. Chociaż istnieją dodatkowe rozszerzenia do X serwera pozwalające rozwiązać ten problem.

Należy też nie zapomnieć o poprawnej konfiguracji bezpieczeństwa serwera. Szczególnie polecam stosowanie specjalnego dodatku do systemu Linux o nazwie GrSecurity (www.grsecurity.org)

Gorąco zachęcam do eksperymentów z X Window System i stosowania rozwiązań terminalowych. Rozwiązanie to może okazać się idealnym do zbudowania taniej pracowni komputerowej czy kawiarenki internetowej.

4. LITERATURA

- [Www.jtz.pl](http://www.jtz.pl) – polska strona tłumaczeń dokumentacji do systemu Linux
- www.ltsp.org – strona projektu budowy bezdyskowych terminali graficznych
- www.xfree86.org – strona darmowej implementacji X Window System
- www.linuxdoc.org – strona z różnymi dokumentami nt. Linuksa
- etherboot.sourceforge.net, www.Rom-O-Matic.net – strony z których można pobrać oprogramowanie do kart sieciowych
- <http://public.planetmirror.com/pub/xdenu/> - strona konkurencyjnego projektu do LTSP, obecnie nierozwijalna
- <http://www.solucorp.qc.ca/xterminals/> - konkurencyjnego projektu do LTSP