

## X11

---

Системите UNIX в миналото са използвали X Windowing System, за предоставяне на конструкциите за графичен интерфейс.

X поддържа екрани, рисуване и преместване на прозорци, както и устройства за въвеждане като мишки и клавиатури.

X Windowing System е създадена от MIT през 1984г.

Версия 11 на протокола е издадена през 1987 г., и оттогава е в тази версия. През 1991 г. беше пусната безплатна версия на X, наречена XFree86.

През 2004 г. XFree86 разпространява код, който е в конфликт с GPL лиценз, така че е раздвоен в X.org.

Повечето дистрибуции на Linux преминаха към X.org, и последната версия на XFree86 беше пусната през 2008 г. Развитието на XFree86 спря.

Когато говоря за X в този урок, приемете, че говоря за системата за прозорци X, XFree86 или X.org.

Архитектурата за всички е една и съща.

X Windows по подразбиране е мрежово осведомен.

Съществува услуга, наречена X сървър

където е видеокартата и входният хардуер.

Това означава, че X сървърът обикновено е на локалната работна станция.

Той обработва входа от клавиатурата и мишката и извежда към видеокартата.

Графичните приложения, които се изпълняват, са X клиенти.

По подразбиране X клиентите говорят през локалната мрежа с X сървъра, използвайки X протокола.

Тъй като архитектурата е изградена от модел клиент-сървър, лесно е да имате отдалечени графични приложения.

Но къде е клиентът и сървърът в тази ситуация е обратното на това, което може да си мислите.

Ако стартирам приложение като Firefox на отдалечен хост, графичния интерфейс се връща при мен на локалната машина, но приложението се изпълнява на отдалечения процесор.

Не забравяйте, че сървърът е мястото, където е входният и видео хардуерът, така че X сървърът е локален за мен, и клиентското приложение е отдалечено.

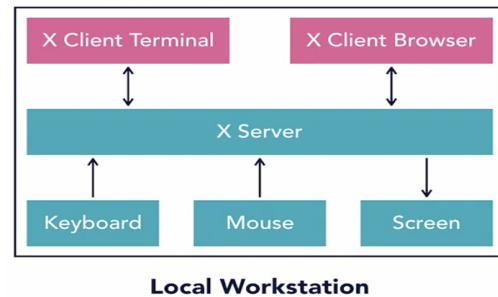
Това е обратното на повечето взаимоотношения клиент-сървър.

X обаче има някои проблеми.

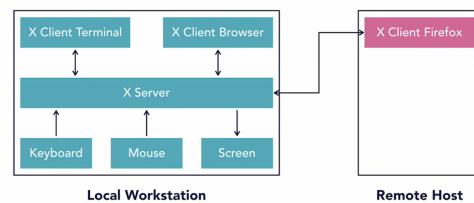
Той трябва да остане обратно съвместим, така че системата стана доста сложна през последните 40 години с множество различни механизми за рендиране, системи за драйвери и петна от код навсякъде.

Като такъв е трудно да се направи бързо.

Освен това нашите нужди са се променили. 3D приложения, като игри, са се оказали по-трудни с X, отколкото с други системи.



Local Workstation



Remote Host

X беше чудесно за сървъри, но тъй като настолните компютри стават все по-често срещани, имахме нужда от нова система. Водещият заместител на X е Wayland. Разработката на Wayland стартира през 2010 г. Първото издание беше през 2012 г. Основяването е по-бавно от очакваното, но повечето дистрибуции предлагат Wayland сега. Wayland се различава от X по това, че архитектурата е по-проста.

[https://en.wikipedia.org/wiki/Wayland\\_\(display\\_server\\_protocol\)](https://en.wikipedia.org/wiki/Wayland_(display_server_protocol)).

( Wayland е комуникационен протокол, който определя комуникацията между дисплейния сървър и неговите клиенти, както и внедряване на C протокол на C библиотека. Дисплейният сървър, използващ протокола Wayland, се нарича компонент Wayland, тъй като допълнителна изпълнява задачата на менеджър на прозорец за композиране. )

В средата няма монолитен сървър.

Не се справя с рисуването.

Той поддържа директно изобразяване от клиента.

Управление на паметта, планиране на команди и настройка на режим са в ядрото.

Други функции се избутват в библиотеките, оставяйки Wayland много по-малък и по-лек. За обратна съвместимост, Wayland поддържа да има X сървър като клиент. И накрая, Wayland в момента няма способност за работа в мрежа, въпреки че се работи по нея. С Linux графичният интерфейс е просто набор от приложения, работещи в операционната система.

Това означава, че можем да променим графичния интерфейс и да използваме друг, ако желаем.

В Linux система, наречена X Windows, предоставя възможността да има GUI, включващ показалеца на мишката, клавиатура и екрани.

На върха на X-Windows ще има GUI Toolkit.

Обикновено GTK или QT.

Мениджърът на прозорци и средата на работния плот използват инструментариума за създаване на прозорци, стартери, ленти с инструменти и менюта.

Също така е важно да се отбележи, че всички приложения работят на всички настолни компютри.

Изборът на един работен плот не изключва използването на приложения от друг.

Приложенията за KDE обаче може да не изглеждат естествени на работния плот на GNOME и обратно.

Има много причини, поради които съществуват различни среди за настолни компютри.

Може би една работна среда използва по-малко памет или друга е предназначена за сензорен интерфейс.

За нас е важно да намерим работната среда, която отговаря на нашите нужди.

Настолните компютри могат да включват и допълнителни приложения, включително текстови процесори, електронни таблици, стартови приложения, инструменти за конфигуриране и др.

В този курс ще използваме CentOS 7 Linux.

Единственият официално поддържан работен плот са GNOME 3 и KDE 4.10.

Въпреки че официално се поддържат само два настолни компютъра, ние можем да инсталирате други среди за настолни компютри, като добавим хранилища на софтуер на трети страни.

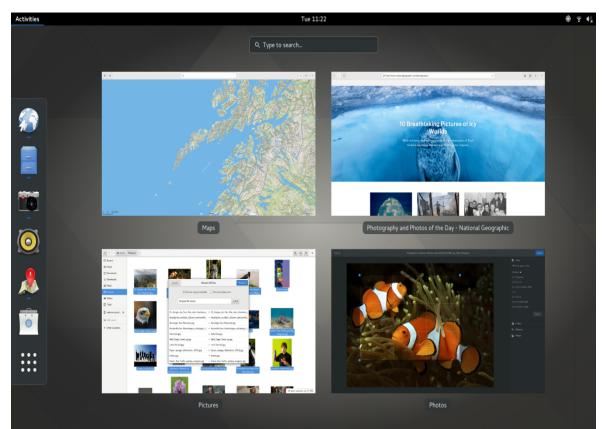
Нека започнем, като говорим за GNOME 3, като работния плот по подразбиране на CentOS 7.

GNOME 3 е част от проекта GNU, стабилен е и има доста уникатен начин на работа.

Режимът за преглед е въведен с GNOME 3 и представлява промяна на парадигмата в работния процес.

GNOME дори не включва меню по приложение по подразбиране, и вместо това разчита на преминаване към режим на преглед и въвеждане на името на приложението.

Той има док-станция за икони за стартиране на общи приложения.



Приложенията на GNOME използват набора от инструменти за прозорци GDK3.

Другият официално поддържан работен плот е KDE Plasma Workspaces, който е по-скоро като Windows среда на работния плот с традиционно меню за приложения.

Някои наричат KDE най-мощният работен плот в Linux и по множество причини.

Той има решение за семантично търсене и дейности.

Дейностите позволяват на потребителя да превключва между конфигурациите бързо.

KDE използва инструментариум QT, изписва се Q-T, като настройва приложенията на KDE като част от повечето други настолни компютри.

Следващите няколко настолни среди не се поддържат официално в CentOS, и като такива трябва да добавим хранилища на софтуер на трети страни, за да ги инсталлираме.

MATE може да е най-популярният настолен компютър за CentOS 7.

Това е така, защото по същество е GNOME 2. Някои потребители не харесаха драстичните промени, направени в GNOME 3, така че разделиха стария код и го преименуваха в MATE.

MATE е по-опростен работен плот с традиционен работен поток. MATE използва инструментариума GTK2.

Един от проблемите с MATE е, че той използва по-старите библиотеки GTK2.

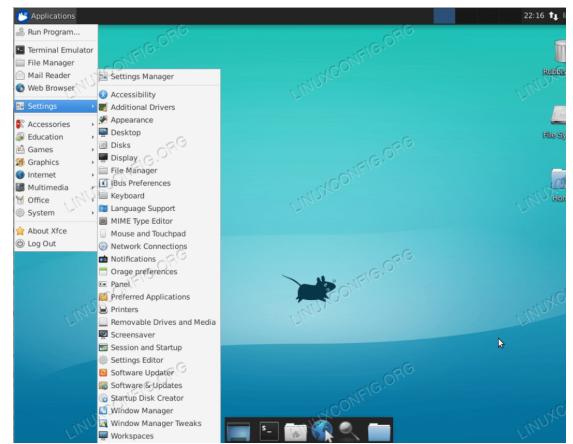
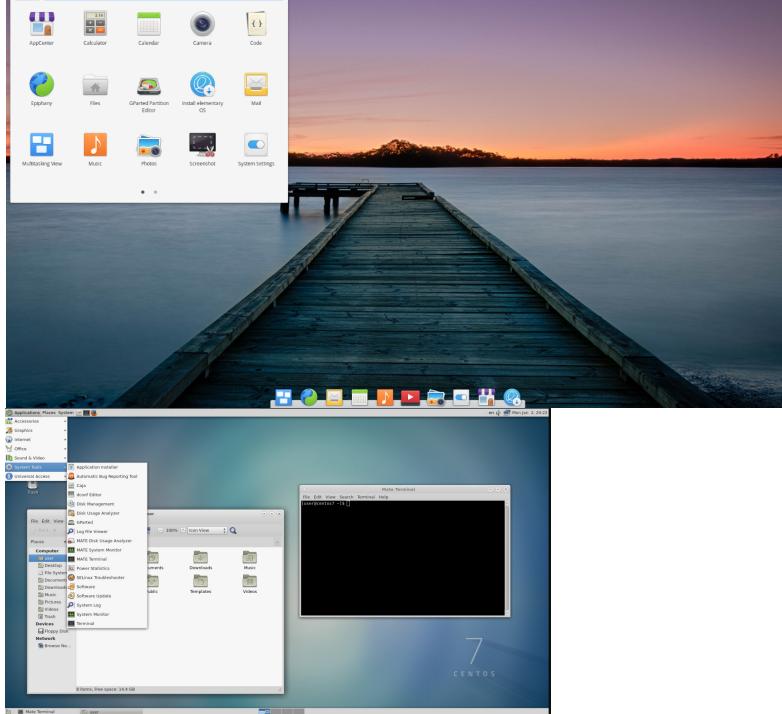
"Някои разработчици бяха загрижени, че тези библиотеки ще останат, така че те се заеха да пресъздадат GNOME 2, използвайки по-нови библиотеки GTK3 от GNOME 3."

Името на този нов работен плот е Cinnamon. Създаден е от Linux Mint Distribution да осигури модерна алтернатива на GNOME 3, но с по-традиционн работен процес включително док за приложения и меню за приложения.

Последната работна среда, която ще разгледаме, е Xfce.

Xfce има традиционен работен процес с меню на приложение.

Основното твърдение за славата е, че използва по-малко памет и цикли на процесора от останалите настолни компютри, които имат повече функции.



Повечето модерни операционни системи, включително Linux, поддържат помощни технологии за потребители със слух, нарушения на речта, зренето или мобилността.

Помощните технологии включват приложения за клавиатурна навигация, еcranни визуални клавиатури, еcranни четци, синтезатори на реч, както и визуални и аудио сигнали.

Може да ги разгледате в Settings --> Universal access.

В миналото с x386 е трябвало да конфигурираме X Windows ръчно чрез редактиране на конфигурационния файл.

Когато Xorg for беше пуснат, това е способност за автоматично конфигуриране на графичния интерфейс и настройките за въвеждане се подобриха и, в много случаи, вече изобщо нямаме конфигурационен файл.  
Все пак можем да го конфигурираме ръчно ако желаете, като създадете файл /etc/X11/xorg.conf.

Има множество раздели за конфигуриране в този файл за обработка на различни настройки, след това различни секции са свързани заедно, за да представляват една пълна конфигурация на устройства.

Файловият раздел се използва за задаване на имена на пътища изисква от сървъра.

Действието ServerFlags се използва за указване на кои глобални опции за сървър Xorg.  
Разделът на модула се използва за определяне на кой Xorg сървърният модул трябва да бъде зареден.

Този раздел се игнорира, когато сървърът Xorg е изграден в статична форма.

Типът модули, които обикновено се зареждат в това раздела са модули за разширение на сървъра Xorg.  
Разделът за разширение се използва за уточняване на кои разширения на протокол X11 трябва да бъдат активирани или деактивирани.

Конфигурационният файл може да има множество секции InputDevice.  
Последните актове служат като HAL на служителите или обратно udev за изброяване на входно устройство и входно горещо включване.

Ако е активирано горещо включване, въвеждане на раздели на устройства с помощта на мишка, клавиатура, и драйверите на VMMouse ще бъдат игнорирани.

Ако горещото включване е деактивирано, ние ръчно ще конфигурираме нашите мишки и клавиатури тук.

Конфигурационният файл може да има множество раздели на входния клас. Разделите се използват за предоставяне на конфигурации за цели класове входни устройства, за разлика от задаването на конкретно устройство.

Можем да намерим съвпадение по продукти, доставчици, драйвери, оформления и т.н.  
Конфигурационният файл може да има множество раздели на устройства.  
Трябва да има поне една за използваната видеокарта.  
Записът на идентификатора за раздела указва уникално име за това графично устройство.

Разделът на водача посочва името на драйвера, който да използвате.  
Раздел за устройство се счита за активен, ако е посочен от активна секция на екрана, до която ще стигнем след малко.

Конфигурационният файл може да има множество раздели на монитора, които описват възможностите на нашите видео монитори включително разделителна способност и честоти на сканиране.  
Конфигурационният файл може да има няколко раздела за режим.

Тези раздели предоставят начин за дефиниране на набори от видео режими независимо от секциите на монитора.  
Конфигурационният файл може да има множество раздели на екрана трябва да има поне един за използванятия екран.  
Екран представлява свързването на графично устройство и монитор.  
Екранната секция се счита за активна, ако е посочена от активна секция ServerLayout.

ServerLayout представлява обвързването на един или повече екрани и едно или повече входни устройства за формиране на пълна конфигурация.  
За да създадем ServerLayout, ще имаме устройство с идентификатор и драйвер.  
Ще комбинираме устройството с монитор, за да оформим екран.

Ще имаме и устройство за въвеждане, което определя мишка и клавиатура.  
След това ще създадем пълна конфигурация

чрез комбиниране на входното устройство с екран  
в ServerLayout.

Въпреки че можем да творим  
ръчно цял конфигурационен файл на Xorg,  
всъщност не се препоръчва.

Всеки раздел, който оставяме извън конфигурацията  
ще бъде автоматично конфигуриран от Xorg.  
По-добра практика е да се заменят части от конфигурацията  
чрез поставяне на парчета от конфигурацията  
в /etc/X11/xorg.conf.d/

Нека да отидем в тази директория и да я изброим.  
В тип терминал в cd / etc / главен регистър X11 / xorg.conf.d /  
и натиснете Enter.  
Въведете LS. Виждаме име на файл 00-keyboard.conf.

Нека го разгледаме с cat,  
въведете cat пространство 00-keyboard.conf и натиснете enter.  
Виждаме това като определяне на входен клас и уточняване  
оформлението на клавиатурата като "US"  
можем да заменим това, като редактираме този файл.  
Можем да заменим всяка от настройките на Xorg  
използвайки същия този метод.

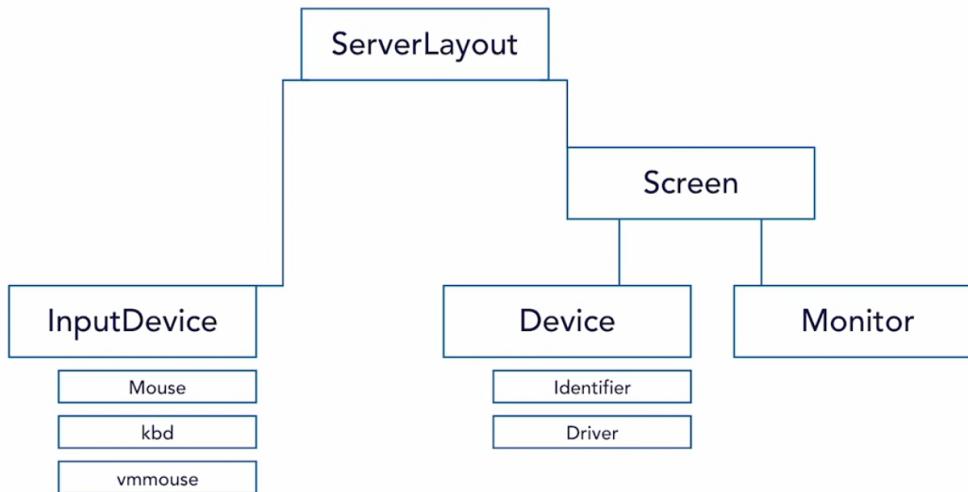
```
# /etc/X11/xorg.conf.d/00-keyboard.conf
```

Както обсъдихме по време на лекцията, ние открихме в  
/usr/share/X11/xorg.conf.d/ конфигурационните файлове за устройства.

```
Section "InputClass"
Identifier "system-keyboard"
MatchIsKeyboard "on"
Option "XkbLayout" "fr"
Option "XkbModel" "pc105"
Option "XkbVariant" "mac"
Option "XkbOptions" ""
EndSection
```

```
# /etc/X11/xorg.conf.d/20-video.conf
Section "Device"
Identifier "Intel Graphics"
Driver "intel"
Option "Backlight" "intel_backlight"
EndSection
```

## Complete Configuration



Операционните системи Linux са проектирани с вграден отдалечен достъп.

Най-често това е чрез командния ред на SSH.(Secure Shell (SSH) )

Има обаче протоколи, които позволяват отдалечено GUI достъп също.

X Windows е де факто стандартът за отдалечен работен плот на Unix и Linux системи плюс има VNC, или Виртуални мрежови изчисления, RDP или протокол за отдалечен работен плот,

протокол, създаден от Microsoft и нов протокол, наречен SPICE.

В повечето случаи се осъществява отдалечен GUI достъп чрез използване на приложение на клиента който се свързва с услуга на отдалечения хост

чрез протокол за отдалечен работен плот.

Отдалеченият хост актуализира дисплея на локалния хост.

След това локалният хост изпраща събития от клавиатура или мишка обратно през мрежата към отдалечения хост.

Отдалеченият хост актуализира базиран на Framebuffer

върху резултатите от действията на мишката и клавиатурата.

X Windows, старата GUI система, използвана от Linux,

поддържа отдалечен работен плот по подразбиране.

X Windows е де факто стандартът за графични настолни компютри на Linux, въпреки че Wayland го заменя.

С X Windows е клиент, който се намира

където приложението работи и сървър

където са видеокартата и входните устройства.

Той обаче работи само когато е инсталиран X Windows

като на клиент, така и на сървър, което в повечето случаи

е Linux или Unix клиент, свързващ се към Linux или Unix сървър.

В локална, не-отдалечена среда,

клиентът и сървърът са на един и същ хост.

X клиентът комуникира с X сървъра

за показване на работния плот и управление на устройства за въвеждане

В мрежова среда Клиентът се движи към отдалечения хост и сървърът се намира на локалния

където са видеокартата и входните устройства.

Това може да изглежда противоположно на това, което бихте очаквали

тъй като сървърите обикновено са отдалечени за нас.

Протоколът X е доста ефективен.

Тъй като X поддържа рисуване на обекти, а не само растерни данни

поддържа обекти като екрани, прозорци,

указатели и въвеждане на мишка и клавиатура.

Той поддържа обектно-базирани актуализации на екрана.

Може да прави цели отдалечени настолни компютри

или просто експортирайте единични приложения.

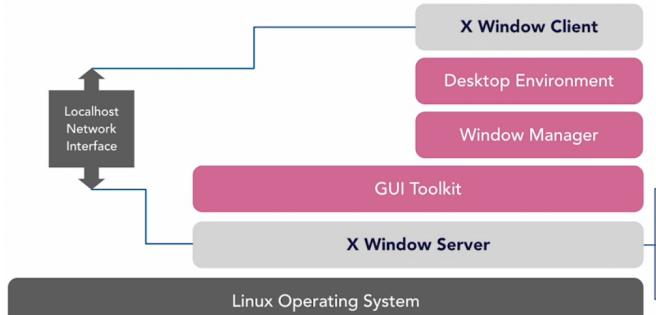
За сигурност може лесно да се тунелира през SSH.

NX, разширение на трета страна, предоставя допълнителни функции

които подобряват X, включително автоматично криптиране

осигурено от SSH, битово компресиране,

X Компресиране на съобщения в Windows, управление на сесии



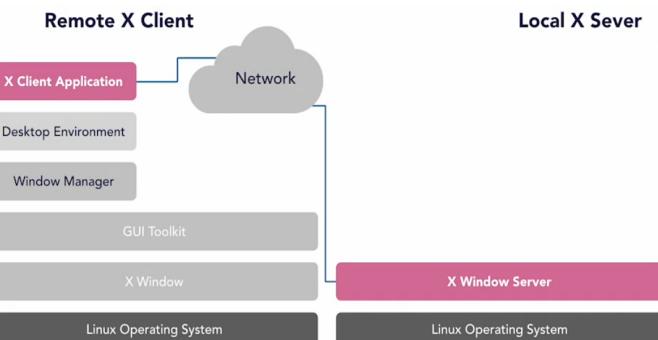
и локално кеширане на данни.  
 Крайният резултат е значително подобрена скорост и използваемост  
 и X обаче не поддържа 3D настолни компютри  
 като GNOME 3 или Unity, който се обажда  
 поставя под въпрос бъдещето му в Linux.

За системи, които не поддържат X Windows  
 нека разгледаме по-отблизо други протоколи за отдалечен работен плот  
 като VNC, RDP и SPICE.

Създаден е VNC, или Виртуална мрежа  
 в изследователската лаборатория Olivetti & Oracle.  
 През 2002 г. е закупен от AT&T и лабораторията е затворена.  
 Разработчиците напуснаха и стартираха RealVNC.  
 VNC използва концепцията за отдалечен Frame Buffer.  
 Той предава събитията от клавиатурата или мишката  
 от един компютър на друг, препредаващ графичния еcran  
 актуализира обратно в другата посока по мрежата.  
 По същество предава снимка на екрана.  
 Като такъв е лесен за изпълнение  
 от гледна точка на програмирането.  
 Протоколът RFB по подразбиране не е защитен  
 и предава пароли в обикновен текст.

Освен това базовият протокол не е много бърз  
 и е доста голяма честотна лента.  
 Пуснати са и други версии на VNC  
 за решаване на някои от тези проблеми като UltraVNC  
 и търговската версия на RealVNC  
 които поддържат криптиране.  
 TightVNC препоръчва използването на SSH или VPN за тунелиране на целия трафик  
 вместо да включва криптиране.  
 TightVNC включва и разширения за компресиране на графиките  
 намаляване на натоварването на честотната лента и подобряване на скоростта.  
 VNC зависи от платформата.  
 Има клиенти и сървъри за много GUI-базирани  
 операционни системи и за Java.  
 Популярните приложения на тази технология включват отдалечени технически  
 поддръжка и достъп до файлове от разстояние.  
 С VNC всяка връзка ще бъде  
 към различен порт на VNC сървър.  
 Ако имахме двама потребители, които искат достъп до настолните компютри от разстояние  
 трябва да стартираме два сървъра на отдалечения хост  
 работи на два различни мрежови порта, по един за всеки потребител.

RDP или протоколът за отдалечен работен плот е собствен протокол  
 разработен от Microsoft за отдалечен работен плот.  
 RDP версия 10 поддържа 32-битово цветово пространство,  
 128-битово криптиране, пренасочване на аудио и файлова система,  
 пренасочване на принтер, сериен и паралелен порт,  
 пренасочване на USB порт, споделена клипборда  
 между локалната и отдалечената системи,



безпроблемни прозорци, в които можете да изпратите прозорец вместо вместо целия работен плот, поддръжка на множество монитори и виртуализирана поддръжка на графичен процесор. RDP клиентите съществуват за много различни операционни системи освен Windows, включително Linux, Unix, macOS, iOS, Android и други.

RDP се наричаше Terminal Services за Windows и сега се нарича Remote Desktop Connection. RDP сървъри съществуват за операционни системи Windows, Unix, macOS. Има и RDP сървър с отворен код наречен xrpd за операционни системи с отворен код.

SPICE е простият протокол за независими изчислителни среди. SPICE е система за показване на отдалечен работен плот създаден за виртуална среда, която позволява на потребителите за да видят компютър

Пример за свързаност при която потребителите използват специфичен порт през който да се оторизират в системата \*портовете ще се обсъждат в бъдещата тема за мрежи



Архитектурен модел на спайс

