



За X-а и хората

Александър Шопов
Сдружение „Свободен Софтуер“

Едва комбинирани чумата и напалмът могат да унищожат европейската софтуерна индустрия по-бързо от софтуерните патенти!



Добри резултати дотук

- Разширението Рендър – нов модел на изобразяване:
 - Операции на Портър и Дъф
- Библиотеките xft2 и fontconfig:
 - Достъп до цялата информация на шрифтовете, избор на шрифт;
- Cairo:
 - Изобразяване на векторни обекти като чертане и запълване на сплайнове, трансформации. Еднакъв изход към X Window (Render), OGL(GLX/AGL), PS, PDF, SVG, PNG.



Последните нововъведения

- DMX – Множество екрани на различни машини да работят заедно, заедно със Xinerama – обединяват се;
- Composite – приложенията изобразяват върху скрити буфери, специален мениджър ги обединява – прозрачност, намаля трафика, опростява.
- Damage – показва, кои части на приложението са променени и трябва да се обновят;
- Xfixes – наблюдение на различни обекти в това число регионите, които описват областите, които са променени;
- XEvIE – прихваща входните събития от входни устройства за достъпност и сложни трансформации.



Промени в обстановката

- Лицензната политика на XFree86 се промени – за културата на X, това не е чуждо явление;
- Нуждата от решение за крайни потребители е голяма;
- Производителите на хардуер започват да се интересуват от нуждите на свободния софтуер;
- Създаде се ново ядро на разработчици към freedesktop.org;
- Целта е да се включат повече участници и да се включи по-добре в културата на Линукс;
- Ново подреждане на хранилището, нов начин на компилиране!



От Xlib към XCL

- Днес почти всички приложения използват Xlib като връзка с протокола X;
 - Над 400 файла, 150 000 реда код даващи споделена библиотека от над 750kB на ГНУ/Линукс за i386;
- Архитектура:
 - Транспортен слой;
 - Протоколен слой;
 - Всичко останало;
 - Стил на писане – да се минимизира броя на извикванията на функциите. Много се използва препроцесора. Трудно се гарантира двоична съвместимост.



Проблеми с Xlib

- Отделните файлове се четат лесно, но трудно се схваща „общата картина“;
- За да проработи добре на компютрите от 80-те, имплементацията е огромна ;
- Не е гъвкава – множество неща са дадени с функции и макроси, трудно се получава двоична съвместимост;
- Вътрешните заглавни файлове се инсталират стандартно, много приложения зависят от „частни“ функции и данни;
- Множество протоколни заявки, които често са излишни, не е лесно използването на нишки. Единствена популярна програма - iso.



ХСВ – ново начало

- Директна връзка с протокола;
 - Функции, които пращат протоколни заявки;
- Много по-малка и проста – при стандартно компилиране $\approx 26\text{kB}$, $\text{Xlib} \geq 750\text{kB}$;
- Не се излиза извън целевата функционалност;
- „С“ не е подходящ за описание на протокола X . Специфичен описателен език за него. Голяма част от кода се генерира автоматично. Преизползване за връзки към други езици, документация, текстово представяне за дебъгери, протоколни връзки от страна на сървъра.



Бъдещето на Xlib

- Да се направи XCL – Xlib Compatibility layer;
 - Първи подход – да се започне от нула като се добавят тези API-та, които реално се използват – неуспешен;
 - Да се започне от текущото дърво на freedesktop.org и да се заменят части.
 - Пълна поддръжка на Xlib;
 - Новата структура на софтуера и компилирането му – от Imake към Autoconf/Automake/Libtool.
 - Транспортния слой – частична миграция;
 - Протоколния – по-успешна. Използва се автоматичното генериране на код.



Промяна на екрана и миграция

- Първоначално в X11, никой не предполага, че екраните ще променят разделителна способност или дълбочина на цвета;
- Проблем при промяната – приложенията чертаят върху повърхност – каква ще е тя:
 - Ще бъде ли хардуерно ускорена?
 - Ще може ли да се прехвърли от единия формат на пикселите в другия?
 - Не е само за един екран – ако искаме да мигрираме или размножим приложение, проблем при безжични връзки.



Какво може да се направи?

- С Render пикселите се преобразуват от едно представяне в друго. Архитектура за общо вътрешно представяне.
- Библиотеките на графичните обекти са правилното място;
- Поддържа ли хардуерът ускорение при ротация?
Хардуерът не позволява да променим начина на четене, затова променяме изображението в буфера. Промяна на размера – промяна на областите за изрязване, промяната на дълбочината на цвета – да се препрограмира хардуера;
- Приложенията трябва да бъдат уведомявани.



Ускоряване, ускоряване, ускоряване...

- Ускорението е много важно, но е различно за 2D и 3D. С 3D се покрива и 2D;
- Облекчават се разработчиците на драйвери и на системен софтуер.
- Не винаги е възможно, трябва да може да се използва и само 2D;
- Трудно е да се интегрират 2D и 3D обектите.



Glitz – OpenGL за Cairo

- Имплементация на изход за най-популярното и поддържано в хардуера API за Cairo.
- Близка до Render;
- Приложение>Cairo>Glitz>OpenGL>GLX>X Window.
- Разлика при изобразяването – точно/неточно;
- Хардуерът е много бърз! В частност много по-бърз и от най-добрите софтуерни имплементации.



Отделяне от хардуера

- Имплементациите на протокола X за свободните операционни системи правят прекалено много работи – управление на хардуер (често лош хардуер);
- Огромна кодова база, която се поддържа трудно, повтаряне на кода, хардуерът се програмира от много различни места;
- 2D в потребителското пространство – не се използват прекъсвания, трудна поддръжка на DMA.



Какво ще се подобри

- По-малко код;
- Драйверите да отидат изцяло в ядрото, само те да програмират хардуера;
- Имплементациите на протокола X Window ще заемат правилното си място в потребителското пространство;
- Ускоряване;
- Възможност да се замени една графична система с друга.



Сигурност на X Window - проблеми

- Надеждна защита на достъпа на съдържанието на прозорците;
- Сигурно отбелязване на прозорци и правилното идентифициране на собственика на всеки прозорец;
- X Window е в потребителското пространство, а не е в това на ядрото \Leftrightarrow SE Linux:
 - Няма контрол върху графични обекти;
 - Приложенията да пращат и получават входа и изхода на другите приложения;
 - Имената на прозорците и иконите се задават от самото приложение – възможност да се излъже потребителя.



Сигурност на X Window - решения

- Съществуващото разширение: „Security Extension“ е недостатъчно и не е популярно;
- На всеки обект в X Window трябва да се даде контекст на сигурност;
- След това се задават политики както в SE Linux;
- Промяна в X Window
 - X Security Module – аналог на Linux Security Module, произволни и коопериращи си разширения за сигурност;
 - Разширение за връзка със SE Linux, нова протоколна заявка, за получаване на контекста на сигурност на основния прозорец на приложенията.