README for XFree86 3.1 on Mach

Robert Baron 17 December 1994

Contents

1 README for XFree86 3.1 on Mach

NOTE: THIS TEXT WILL LOOK A LOT LIKE THE README.MACH FOR XFREE86 2.1, BECAUSE NOT MUCH HAS CHANGED FROM THE MACH OS PERSPECTIVE. ON THE OTHER HAND, THERE ARE SIGNIFICANT DIFFERENCES IN THE BUILD INSTRUCTIONS. BE CAREFUL TO FOLLOW THEM.

All XFree86 client binaries should work with not only the Mach 2.5 kernel but also the Mach 3.0 kernel. Mach 2.5 kernels should support both the traditional SVGA/VGA servers and the newer accelerator servers. The Mach 3.0 kernel MK83 has a fix necessary for Mach 3.0 to support the new accelerator servers and for SuperProbe; this change allows the full 64K of I/O ports to be accessed. **THUS, IF YOU USE MACH 3.0, YOU MUST RUN MACH 3.0 Kernel RELEASE MK83 OR LATER TO USE ANY ACCELERATOR SERVER**. The XF86_SVGA server, XF86_VGA16 server, and XF86_Mono server should still continue to function under the older Mach 3.0 kernels. Note: the latest SuperProbe can be used with older Mach 3.0 kernels by specifying the command line option `-no16`.

Autodetection of SVGA clocks will not work at all with the Mach 3.0 kernel; the clocks must be specified in the **`XF86Config`** file. Mach 2.5 kernels can compute the clocks correctly. (So you might need to use Mach 2.5 to get the data for the clock line.) Autodetection of the SVGA chipset requires Mach 2.5 release X142E or later, or Mach 3.0 release MK76 or later. Alternatively, you can specify a Chipset line in the **`XF86Config file`**.

Lastly, only the ATI Mach32 Accelerator server (**`XF86_Mach32`**) has been extensively tested under Mach 3.0. It works reasonably and the OS intensive features, like linear memory $non-bankswitched$, seem to work fine. An S3 (801) board on an ISA bus has been tested and also seems to work. We have not checked the non-banked memory feature on S3 or other cards.

2 USING AND CONFIGURING THE INSTALLED BINARIES

By default, the CMU BINARY DISTRIBUTION obtained via sup is installed in `/usr/X11R6` make sure you have enough room for it (about 45Meg). The previous server for XFree86 2.1 was installed in `/usr/misc/.X11`; thus you can easily keep both servers around.

Mouse selection for Mach is different from that described in the `XF86Config` manual entry. The Pointer Device entry in `XF86Config` is ignored. The mouse device is always `/dev/mouse`.

Mach supports several kinds of external mice, the logitech three button mouse, the microsoft two button mouse, the PS2 mouse, and the logitech mouseman mouse. We do not support any kind of bus mouse. The way you indicate the mouse type and com port is with the mouse device minor. The value you set the minor to is computed by the formula:

$<$ type $>$ * 8 + $<$ com port $>$

$<$ com port $>$ is 0, or 1 or … depending on which com port you plug your mouse into. (Some architectures prefer to make com0 the modem line.)

$<$ type $>$ is:0 for logitech mouse
1 for microsoft mouse
2 for ps2 mouse
3 for no real mouse at all
4 for logitech mouse man (M+ or V type)

We have implemented a neat hack for notebooks so that they don't have to be saddled with a HUGE external mouse. Scroll lock toggles a mode wherein "seldom used 'function keys'" are used to represent mouse actions. F1, F2, F3 are the left, middle, and right mouse. The first key stroke represents a down click and the second key stroke represents an up click for the mouse button. Then the up down, right, left, arrow keys actually move the mouse AND if you have a standard pad with home, pageup, end, and pagedown filling in the square around the up, down, right, and left arrow keys then the former set do the obvious and move the mouse along the diagonals. Try it … Note: whereas most notebooks these days do have an integral mouse, they typically provide only two buttons. Thus this feature still is useful for just using F1, F2, and F3 for accessing three mouse buttons.

3 ACKNOWLEDGEMENT

Elliot Dresselhaus did the initial work of converting the SCO only driver to work under Mach2.5. **Ernest Hua** added support for Mach's unique way of treating the mouse. **Joel Jacobson** tested the release to make sure it built at sites other than CMU. **Robert V. Baron** was responsible for Mach3.0 support and feeding the Mach changes back to the XFree86 development group. Lastly, we are indebted to all the **developers of XFree86**.

4 BUILDING THE DISTRIBUTION

Refer to `xc/programs/Xserver/hw/xfree86/INSTALL` for details about building and configuring the distribution.

I build XFree86 under a Tahoe Binary Environment with a Mach 2.5 kernel and I make sure that the Mach 3.0 "release" include files and libraries are not on my path. The Mach interface (API) is slightly different between the Mach 2.5 kernel and Mach 3.0 kernel, but the Mach 3.0 kernel was designed to be upwardly compatible with Mach 2.5 kernel. This means that if you build XFree86 with Mach 2.5 includes and libraries $thedefaultintheTahoeEnvironment$ the binaries will run on either Mach 3.0 or Mach 2.5.

If you choose to build with the Mach 3.0 headers and libraries, the applications will only work with the Mach 3.0 kernel. The Mach kernel calls made in some of the `vga*/*/*driver.c` and SuperProbe use the Mach 2.5 calls names. To build with Mach 3.0 these latter few Mach 2.5 dependencies must be changed.

There are several different Mach Operating Systems available. One is distributed by Mt.Xinu. The others are distributed by CMU; CMU distributes two different OS's; Mach 2.5 - a monolithic kernel and Mach 3.0 a micro kernel with Unix as a server. Both OS's from CMU use the same set of binaries which were developed from the BSD4.3 Tahoe release. (Note the Mt.Xinu system uses a later "generation" of application programs. Lastly, the environment used internally at CMU is still different from what is distributed externally. This environment supports AFS and Kerberos and ...) The CMU Tahoe environment uses a gcc 1.36 compiler with a bsd derived cpp (not __STDC__); the Mt.Xinu system uses gcc 1.37.

We do not build XFree86 under all these systems, just one – CMU Tahoe environment distributed with Mach 2.5. It should be possible to build XFree86 using any application environment and with either Mach 2.5 or Mach 3.0. Please let us know of any difficulties.

```
>
++Some changes are necessary to the Original Tahoe environment to build++
++X11R6.  A few "new" functions are called and were thus added to      ++
++/lib/libc.a(mbtowc.c, putenv.c, strdup.c, and strtol.c) and a few old++
++ header files were modified to guard against multiple inclusions     ++
++(/usr/include/{time.h,utmp.h}).  These three files should be         ++
++dated May 31 1994; if they are not, "sup" to get the latest copies.  ++
++The X11R6 build was done with the gcc 2.3.3 compiler that is shipped ++
++with Mach 3.0.  You probably want to use this gcc, or maybe even get ++
++a later version.  THERE IS ONE DEPENDENCY ON THE COMPILER IN         ++
++xc/config/cf/mach.cf; CPP IS SPECIFIED AS:                           ++
++#define CppCmd        /usr/mach3/lib/gcc-lib/i386-mach/2.3.3/cpp     ++
++#define PreProcessCmd /usr/mach3/lib/gcc-lib/i386-mach/2.3.3/cpp     ++
++THIS ASSUMES THAT THE COMPILER AND MACH3 "RELEASE" ARE ROOTED AT      ++
++/usr/mach3.  IF YOU DO NOT FOLLOW THIS PRACTICE OR USE ANOTHER       ++
++COMPILER, YOU MUST CHANGE THESE TWO LINES ACCORDINGLY.  Before you   ++
++start the compilation you must type:                                 ++
++      limit datasize unlimited                                       ++
++Two of the .c files (lib/X11/StrKeysym.c and                         ++
++programs/Xserver/XIE/mixie/fax/faxtabs.c) generate intermediaries    ++
++that exceed the default limit of 6Meg for data.  Lastly, if you are  ++
++building the Local CMU Environment make sure that /usr/cs/include is ++
```

```
++NOT on your CPATH.                                                  ++
```

The normal XFree86 distribution makes all the servers in bin/ setuid root; this is not necessary. In Mach, access to I/O privileges is not controlled by root; rather it is controlled by `/dev/iopl`. What is necessary for Mach is that the X servers be able to open `/dev/iopl` for read/write. There are many ways to achieve this effect.

`$XConsortium: Mach.sgml,v 1.3 95/01/27 16:14:25 kaleb Exp $`
`$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/Mach.sgml,v 3.4 1995/01/28 16:01:49 dawes Exp $`