

## Re: OT: my new PC rocks!!

**Source:** <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2003-12/0725.html>

---

**From:** Beth (*BethStone21\_at\_hotmail.NOSPICEDHAM.com*)

**Date:** 12/30/03

Date: Tue, 30 Dec 2003 15:55:20 -0000

Frank Kotler wrote:

> *Beth wrote:*

> > *Randy wrote:*

>

> > > *Windows is the only OS I can currently install on the machine as the*

> > > *hardware is too new for most of the other OSes I've got (no, I'm not*

> > > *interested in installing drivers by patches, recompiling the kernel,*

> > *etc.).*

>

> > *Mind you, I \_still\_ can't get Linux installed on my "Linux box" second*

> > *machine...*

>

> > *I can get Linux installed, no problem. (my "first machine" is made of*

> > *spare parts, but I guess they're sufficiently "standard"...)*

Aaarrggh! Will people stop saying that!! I mean, when someone comes up to you and says "I'm not feeling too good, I think I've got a bug and am low and lacking in energy and feel miserable", is the best way to comfort them to jump up and down in Teletubby "happy, happy, joy, joy" Utopia, nah-nah-nah-nah-ing how everything is brilliant in your universe and you're not ill at all? Know what I mean?

Yes, I \_know\_ Linux can install no problems...it \_used to\_ do that for me too...but when it stops doing that, then you'll also go around cursing those "evil" ISO images and all those people smiling like the Cheshire cat in their happy, singing "Toon town" universe of pure joy...

> *Getting Xwindows configured is another story! I don't have the original manual*

> *for my monitor (secondhand - my last monitor literally "fell off a truck"!)* - *maybe if I googled around I could find specs online, but...*

Re: OT: my new PC rocks!!

I'm alright with this because I kept the big monitor box (because it's nice and big :) for storing things in...requires climbing into the attic with a torch but the specifications are nicely written in big letters on the side of the box to type into "Xconfigurator" directly for perfect monitor perfection :)

> *Windows doesn't ask me all those tough questions, why should Xwindows have to? I assume the answer is "better hardware probing". Maybe, just maybe, assembly language programmers ought to be doing something about this instead of complaining about it.*

Well, there may be a bit of "better hardware probing" in the sense that Windows may recognise "plug and play" monitors better...but, really, I don't think the problem is that at all...

You know what the real difference here is? *\_Attitude\_*...think about it...when you boot up Windows, it's usually set to the lowest resolution with a reasonable amount of colours and the refresh rate at one of its lower settings...then, you go into "display properties", move the sliders up a notch, press "test" and see if the screen goes all wobbly (and, yup, it sure can do that :)...if you don't press "yes" in 15 seconds, it reverts back to "safe"...well, try moving the notches to less extreme settings...then try again...once you've got it right, you never go back...and, hey, you might even forget that you spent that time playing with the settings to get them right...

Windows determines the settings "empirically" with the "interactive assistance" of the user...it starts "simple" and then the adventurous user moves the notches higher...X-Windows, though, wants to know the *\_exact numeric values\_* before proceeding...its "attitude" is to use those numbers to work out what's possible with formulas and stuff...

The difference is merely "attitude"; If X-Windows simply chose "640x480x256" and then featured a set of GUI sliders where you could "notch" it up to higher settings, complete with the "if you don't say 'yes' in 15 seconds, then we'll presume the image went all 'wobbly' and it's not a valid setting" thing then you'd find no difference between Windows and X-Windows on that score...

It's two different solutions to the same problem; And though X-Windows' solution is *\_logically\_* "superior" because it shouldn't even offer the choices of settings that'll never work (presuming you give it the correct numbers :), the Windows version is *\_physically\_* "superior" because it has "hidden" those questions behind moving notches and a simple "yes or no" as to whether you can actually see anything on the screen or not...well, any "fool" can move a notch and then say "yes" when it works and "no" (or just leave it 15 seconds, if the screen is so mangled that there's no way you can tell where the

mouse pointer is to move it to the right button ;)...

Windows takes a different attitude: "Hey, those modern monitors can handle being given wrong settings for 15 seconds or so...let the user find the best setting that actually works by simple trial and error"...my video card can do an awful lot more resolutions and refresh rates than my monitor can handle but I know I've got the best resolution that it can handle because I moved the notches up one by one until they were as high as they could go without the image going all "wobbly"...

X-Windows really needs one simple-ish addition (which was originally absent from Windows, until Microsoft wised up :)...the ability to "reset" the resolution / colour depth / refresh rate settings on-the-fly...that is, change the video settings without needing to completely shut down X-Windows and reboot it...as I say, in Windows 3.x days, you also had to select your video settings and driver then shutdown Windows completely and reboot it to see if those settings would work...that was as much a nightmare to use as X-Windows still is to configure...

But if there was some "SetVideoSettings(width, height, colour depth, refresh rate)" API that immediately changed video modes – resetting the desktop to this new resolution – then it would actually be pretty easy to write the "notches" GUI interface that allows the user to test out each setting to see if it works...then, if they click "apply", write the settings to the usual configuration file...

Plus, for "full-screen" computer games and so forth, X-Windows could do with stealing a little of that Windows functionality with some "SetVideoMode" API thingy...as I say, Windows was itself once exactly like X-Windows on this that it was sort of "stuck" in one resolution that wasn't easily changed...but Microsoft added the required video mode changing API and then adding the "Display Properties" utility is child's play after that...

Windows doesn't actually do a better job when you really consider things...what it does, though, is "hide" the fact that you're even doing these things...I mean, you must have played with "Display Properties" sometime but you do it once and it's so easily done, you've not even realised that, effectively, you DID answer similar questions with Windows too...the trick was changing the questions from "please enter the vertical sync rates" into sliding some notches instead...and, sure, we allow for invalid settings to be chosen but we only keep them when the user says "yes, please keep them"...

Microsoft, to their credit, have – though only through "trial and error" from a load of user complaints, I'm sure – learnt that sometimes the "techno-fix" is the worst possible way to fix a problem...sometimes, even if not quite so "impressive", a simpler solution ends up better because it's easier on the user and the

"drawbacks" really are completely tolerable (yes, Windows does allow the image to "go wobbly" but that's not really such a terrible thing to get quite so worried about avoiding it...modern monitors – thanks to having to handle that Windows takes this approach, in fact – will be able to handle bad settings for a short period of time without any great trouble...some, in fact, will simply automatically switch themselves off when given bad settings so you can't cause any damage :).)

> > ...I checked the jumper settings and they  
> > are okay...  
>  
> I assume you're talking about "master/slave" settings? (pardon the  
> politically incorrect language) That's a good start. I've had "bad  
> jumpers", too, that appeared to be in the right place but weren't  
making  
> a connection. "Wiggle the cables" is a good high-tech fix for many  
ills :)

Well, the "jumpers" are correct...that much I've made sure of...and I've connected and re-connected the wires a few times (taking things out, putting things back in, etc. :) that, though I didn't specifically try "wiggle the cables", it would be unlikely that I'd repeatedly make exactly the same mistake each time...anyway, when I plug in a cable, I usually naturally "wiggle" them a bit to make sure they went in correctly...

But it's a very good suggestion...Murphy's law does often tend to rule in these cases and something that simple is often the problem...but not in this case, I'm sure the jumpers and "cable wiggling" is exactly right...I don't take offence at the "easy" suggestions because I've had problems plenty of times that ended up as: forgot to plug it in, loose wires or something equally obvious and silly...

In fact, I had a problem with a floppy drive once...it kept having "bad sectors" at a certain area of the disk...I couldn't work out what was wrong...so, eventually, I phoned the "helpline" and the bloke on the other end simply asked "are you putting the monitor on top of the box?" (it was a pre-tower PC and, as was customary, I put the monitor on top of the "base" :)...I said "yeah, sure"...then he said "take it off...that model's case isn't designed for the weight of a monitor" / "Oh"...I took the monitor off and everything was perfectly fine once again...thinking it through in my mind, those "bad sectors" would be more or less located exactly where the corner of the monitor would have been pressing down on the floppy drive...

Yup, it was that simple...though it was somewhat "traditional" with that style of PC case to simply put the monitor on top of it to save desk space, this particular hardware manufacturer apparently disagreed with this concept...and, nope, it wasn't designed to handle it at

all...I put the monitor in the wrong place...and that was all that was wrong...but, as the problems were all with the floppy drive, I wasn't even looking in that direction at all...the bloke on the helpline knew it straight away, though, the second I quoted the model number because, apparently, *\_everyone\_* had assumed putting the monitor on top was a good idea and this was "complaint #1" when it came to that model...you can't really call it a "design fault" totally because they did know but *\_deliberately chose\_* to make it this way...it's more an "incompatibility" with what users wanted and expected with what some designer decided "should be the case"...

> > *but it still crashes in a recursive kernel panic when I*  
> > *select the "test the CD" integrity test thingy...*  
>  
> *What do we know about this CD? Does a Windows "rescue disk" find it*  
> *okay? I'm using Slackware – they provide several pre–built kernels*  
*with*  
> *different support included. If RedHat does it the same way, maybe a*  
> *different kernel would help?*

Well, no, Red Hat seem to do it completely differently...at least, there's only one package available for free download...which, of course, is how you can say "Red Hat 8" and "Red Hat 9" and so forth because they concentrate on creating a specific "package"...plus, the "test the CD" thing I mentioned is actually part of the installer program and this auto–boots up off the CD with no problems...before starting the install, the installer has the option of quickly verifying the integrity of the three CDs to check that the ISOs were written properly and so forth...much better test it quickly there than to discover the problem half–way through the actual install, right?

Also, they provide MD5 sums for each of the ISOs, which you can use to check they downloaded okay and I did check those and the actual images I downloaded are correct...it might have badly written to the actual CD but then I've never had problems writing to any other CD from the same box and I even selected a slower speed for writing just to make sure it would write okay...it *\_could\_* be that, perhaps, but it doesn't seem too likely as I took all the expected steps to ensure that the CD images were correct...

Plus, what happens when it goes wrong, as I say, is that there's a never–ending stream of text flying down the screen, listing CPU register contents making it look like an exception or something...and this is totally unreadable at the speed it's going and ruins the user interface thing as it scrolls off the top that I'm sure this isn't *\_supposed\_* to happen...the program itself has crashed (because it gets stuck spitting out register contents and exception numbers forever to the screen) rather than this being an "integrity compromised" error message or something...

And if I choose to "skip" the integrity test then when it comes time to start writing things to the hard drive, the program freezes for a second before it automatically reboots itself...which is interesting because nothing goes wrong reading from the CD for the rest of the installer (which uses an X-Windows interface to have a nice, user-friendly install program :)...it's only when it seems to be attempting to begin to access the hard drives that things go wrong...I'm almost certain the actual install CD is just fine because the MD5 sum is right, I took steps to make sure it wrote to the CD correctly and, with Red Hat, it's an auto-booting CD that starts up Linux, starts up X-Windows and runs a fancy installation program all off of the CD and that works 100% fine \_until\_ it tries to go to the hard drive...

> > *and if I try skipping*  
> > *that, then the machine hangs and reboots itself when trying to format*  
> > *the partitions ready for install...*  
>  
> *Some particular partition it has trouble with? Would resizing them so*  
> *they fall on different borders help?*

Now, there's an idea...I have split up the hard drive into partitions because I wanted to leave some room for also installing other stuff...perhaps I should reformat and shuffle those partitions about...but, again, I wouldn't think it was a problem because I had something like that before and there wasn't any trouble (plus, Linux \_is\_ the first partition because I read about some often reported problem with larger hard drives because the BIOS only has CHS or something...so I followed the "read me" advice there that this \_shouldn't\_ be the problem and the BIOS \_does\_ seem to understand such things because the BIOS set-up program correctly reports the drive and lists them as "LBA" / "CHS" and so forth...it's not that old a machine for it not to be able to understand such things in the BIOS ;)...

> > *that's Red Hat, by the*  
> > *way...*  
>  
> *A long time ago, I heard that if you've got a friend running some distro*  
> *of Linux, that's the best distro for you. I didn't really have a friend*  
> *running Linux, but I was talking with the son of an old high school*  
> *friend, and he was using Slackware, so that's what I picked. Never had*  
> *occasion to use another, so I don't know how much difference there is. I*  
> *watched a friend install... I think it was Mandrake – seemed to have a*  
> *fairly "Windows-like" installer... seemed to go well, but the*

network

> *card didn't work. I think I could have fixed it, but she just overwrote*

> *it with something else. Looked like a pretty easy install process... if*

> *it had only caught the network card...*

Red Hat has a "Windows-like" installer too...plus, as I said, I had this distro (the previous version of the package, though ;) installed on this machine before...I was only "upgrading" it to a newer version (but doing so by re-installing it from scratch, as I wanted to also shuffle the partitions at the same time :)...I mean, if this continues not to work, then I might have to simply consider using another distro...but I was happy enough to stick with Red Hat because they work in "packages" and have "Windows-like" installers and so forth that it's \_usually\_ all very nice and simple...you know, stick in the CD, answer some questions about your keyboard layout, time zone and what packages you want then it basically does everything else for you...a fantastic little install program \_WHEN\_ it actually bloody works...my problem here, of course, is that it's not working...and then things aren't quite so happy or rosey with Red Hat...

> *One of my first installs, the network card didn't work. I determined that it was using the right "tulip" driver. Went to some nasa.gov site*

> *that had the newest versions of the drivers (using Windows, which may be*

> *considered "cheating"). Fortunately, the file included the command-line*

> *to build it – I never would have figured it out by trial and error.*

To

> *my astonishment, it worked! "Look ma, I'm a Unix Guru!" But it was just*

> *beginner's luck – haven't had any great success since...*

Actually, here's a tale that'll please the Linux people...because having two machines next to each other, as I've set up here, I thought it would be a great idea to pop some network cards in both and hook them together...making it easy to move files back and forth, allowing both machines to "share" the internet and that sort of thing...like your experience, I got the latest Linux driver off some website, typed in the command-lines and it was up and running problem-free...great...but, in fact, the problems started when I plugged the other network card (same exact card type) into Windows and got stuck in one of those never-ending "plug and play" loops: "you have new hardware" / it installs the drivers off the drivers disk / the driver doesn't seem to be working because it again reports "you have new hardware" / it wants to re-install the same driver / and again and again..."okay", I think, "time for Windows update"...yup, sure enough, there's a newer driver that reportedly fixes bugs (though the bugs listed don't seem to be the one I was having)...I try it with

that instead...nope, still get the "you have new hardware"  
never-ending loop...

I tried everything but Windows simply wouldn't have it...so, regards the "Windows always installs things easily", this isn't completely true because, in fact, for this network card, Linux is the one that's entirely problem-free by just following the instructions for compiling it supplied in a "read me" file...while Windows – with all its "plug and play" stuff – simply couldn't cope and, worse, actually gets forever stuck in a "found new hardware" infinite loop that could really cause serious problems because the only way to actually get out of it is to shutdown and remove the card...which is hardly "user-friendly" to clueless newbie users...in fact, for any writing a hobby OS with some Windows-like "plug and play" nonsense, there's a good suggestion for you: have the OS \_record\_ what it's trying to do and if it senses that it's tried to install the same piece of hardware five times in a row, then simply mark it as "not working" and \_stop\_ trying to constantly "found new hardware" with it...clearly, it's not working and you've got to have an OS that knows when to give up...it can be told to retry the hardware manually or something when the user \_really\_ does want to "give it one more shot"...but when it simply gets stuck in a never-ending loop, that's a poor show...

> > *downloaded off their site...*

>

> *I suppose you mean an iso image? With Slackware, at least, you can also*

> *get "disk sets" – intended to be installed from floppy, but you can put*

> *'em on a FAT partition and install from there. Might be an option if you*

> *can't get the CD working. I guess you can install from an iso image on*

> *hard disk too – I don't know how to do that one.*

Well, Red Hat use "MD5 sums" on all of the ISO images for you to manually check if they downloaded correctly and, presuming that the MD5 sum algorithm is working well, then the ISO images I've got should be 100% fine...

Regards the downloading, you can \_buy\_ box sets of Red Hat...but you can download it for free from the website (that "clause" in the Linux licence about not being allowed to charge for Linux itself but only for "reasonable costs" of media, "extras" supplied and so forth...plus, the source code should also be made fully available and that's in the same FTP download area...hence, to properly comply with the GPL and Linux's version of it, then all distros \_should\_ be offering some sort of similar "it's always possible to get it free, if you don't mind the inconvenience of many hours of downloading" thing :)...I'm going for the download option not just because it's free but also because that actually is the most convenient way...

- > *The very first thing I installed was "ZipSlack" – a very minimal distro*
- > *that installs in a dos directory (or on a "ZipDisk", hence the name). A*
- > *handy way to "ease" into Linux without having to repartition anything or*
- > *uninstall anything. I keep it around to use as a fancy "rescue disk" –*
- > *it's proved useful a couple times. I suppose other distros have a*
- > *similar scheme(?). I doubt if it'd work on NT filesystems, but if you're*
- > *going to be installing '98 anyway... Something like that might help you*
- > *figure out what's going on with the "problem" drive(s).*

Well, yes...and no...because the point here is that this machine is specifically put together as a "second machine" for running Linux (Windows 98 may also go on there just to have a different version of Windows for checking "compatibility" for any programs I write with the two types of 9x / NT kernel...on the other hand, I might put "pure DOS" on there or something else instead, as that could be handy for testing some of the DOS code that gets posted up around here, as the Windows "DOS box" isn't always a perfect emulation ;)...

The general idea being that I'm specifically putting together this "second machine" as a "don't care / experiment" thing...hence, as useful as those kinds of things are for certain conditions, that's not really what I want here...I'm perfectly happy to re-partition everything...in a sense, I want to do so...to have proper working "typical" versions of these OSES for "testing" purposes (hence, the more "normal" it is with the "typical" sort of install, the better it is for "testing" things out on ;)...

Note that I do have "Cygwin" on my Windows machine already, which is a similar thing...it's actually a novel and clever idea...there's a "Cygwin DLL" file that provides all the Linux kernel functions and then just "emulates" Linux under Windows (I guess the "INT 80h" interface is out with this solution but you can always change those to calls to the C interface equivalents :)...they've even converted X-Windows over to a "Cygwin" version that'll run in a window under Windows...but, though it's useful for running GCC and command-line things, the "Cygwin" stuff is just too slow and cumbersome and complicated to write X-Windows programs with (plus, it only has the most basic X-Windows and no GNOME or KDE or anything)...for just running Linux programs on Windows, it's actually a pretty clever idea...but actually developing things for Linux this way is, in my opinion, a bit too awkward and complicated...especially as I'm looking to do X-Windows stuff with ASM, which really doesn't fit too well with what Cygwin can currently manage (it's stretching itself to run X-Windows "under emulation", as you might imagine, that trying to push it further is too much of a "constant struggle" for it to be a good

method of development...not least because, of course, `_IF_` I could only get things to install on that "second machine", I'd have the completely reverse in a development method that's so incredibly easy and convenient...as I think I mentioned before, `_once_` you try cross-development like that, it's sooo easy and convenient compared to anything else, you just can't really tolerate doing it any other way...with two machines, for example, I can use my favourite Windows editors and tools, listening to MP3s and so forth...then simply stick it onto a disk, pop it into the Linux machine and run it there to test if it works...also, if anything is taking its time on the Windows machine, then there's always the option of playing cards on the Linux machine while you wait...for cross-development, there simply is no better way (although, some people may, of course, prefer to use Linux and then merely "test" things on Windows...the other way around...that's sort of the point of this method...it's very flexible and versatile and simply let you use `_either_` machine – even at the same time – to get the job done...where Linux tends to do a better and easier job of things then I can simply use the Linux box...plus, `_IF_` I'd only been able to get that network card working under Windows too, then things would have been even easier still, creating things like "shared folders" between the two that I don't even really need to think about transferring data across :)...

> > ... and I was coming along great with the X–Windows stuff I was  
> > doing...  
>  
> Yeah! Been hoping for some examples from you.

And they would have turned up by now...like I wanted to provide a simple example of a basic "hello, world!" for that poster who was asking about X–Windows before...but, basically, until I can get my "Linux box" to work, I can't test or properly develop the code...that's what's most frustrating about this "install problem"...it's just totally ground all my plans to a halt until I can work out what's wrong to fix it...

> I've seen a couple  
> examples from `Numi_tor` – one using `xlib` directly and one using the Gnome  
> Toolkit – they both look pretty complicated – even more "housekeeping"  
> than Windows to do before you can do anything.

Odd; With `XLib` directly and the most `_minimal_` "hello, world!", I always liked X–Windows because it had `_less_` nonsense involved and far more flexibility...again, I'd demonstrate this with some code but, without the Linux machine, I can't really test it (plus, though I'm so used to writing Windows code that I can put together a "Hello, world!" blind, so to speak...I've done less X–Windows (and Linux too) stuff that I'd still like to have access to "man" and "info" pages and references too :)...

But, anyway, talking about it instead of example code, I found X-Windows to be better on a number of scores...at least, though it may not always look as pretty or be as responsive of Windows, the underlying X-Windows design has a better style to it...Windows "wins" on "fancy graphics" because Microsoft know the old trick of "as long as it looks good on the screen, most users are fooled by this into thinking its a better product...when, in fact, 'under the hood' it can be a hundred times worse, as long as it looks a hundred times better on the screen, it'll surely sell like hot cakes"...hence, X-Windows is a bit complicated to understand when you first read the documentation because it's going on about "bitplanes" and stuff, where GDI doesn't mention these...but, of course, what has to be remembered about X-Windows is that it works on a wide, wide range of computers and, subsequently, hardware that this stuff does make plenty of sense ("bitplanes" were THE standard way of doing graphics, right up until "mode 13h" on the PC introduced the simplicity of "one byte per pixel" and DOOM was written for it and it changed how things worked from then on...every machine – including PCs – were "bitplaned" right up until this "cheap hack" for getting a 256 colour mode on hardware that only had 4 bitplanes (16 colours) hard-wired from the EGA days...as is quite typical, this industry has been ruled by "happy coincidences" and "accidental empires" all over the place...because, in fact, when you had a low (16, 8, 4 colour displays :) amount of colours, bitplanes can actually be superior...more complicated but, as can be done in VGA mode 12h for example, multiple pixels all written at the same time...in fact, much like "video killed the radio star", DOOM forever killed bitplanes...not only did it use mode 13h's then-unique method to very good effect, it made the 3D textured polygon "king"...and bitplanes aren't at all useful when you have a high number of colours with every pixel in a scanline a different colour (typical of texture mapped polygons :)...actually, it was Wolfstein that first picked up the knife to kill bitplanes but DOOM knocked everyone's socks off in such a permanent way that that was the day they truly died...now merely a "historical curiosity"...only bothered to be used for VGA's "mode 12h" for some sort of "safe mode" with reasonable resolution (because beyond mode 12h, you have to start going VESA VBE and device drivers and DirectX / OpenGL...the entire landscape changes once you step beyond 640 x 480 x 16 colours ;)...

Anyway, this "bitplanes" thing can certainly look confusing and put people off X-Windows because Windows does look at graphics in a more simplified way (but, then again, that's a trade-off...if you know what you're doing, then "doing it yourself" very often gets better results...Windows has simple "generic" libraries to do the hard work and all the usual arguments of "DIY versus libraries" apply...although, with "accelerators", Windows wins an edge by using that hardware capacity to "patch over" any "overhead" the library stuff may bring...meaning that Windows does tend to do better but only because the hardware is doing all the "grunt work"...a similarly wised-up "accelerator-aware" set of X-Windows API would catch-up and theoretically exceed because of the more "low-level"

perspective...X-Windows is only really "behind" on such things because it's designed to work on any hardware – and does do so – and has to take a "generic" perspective in its API...it may be hard to appreciate when you don't see X-Windows perform better on the screen because Windows is exploiting all the PC hardware, but X-Windows really does – on many scores – have a much better underlying design to it :)..

Actually, X-Windows lacks the "overhead" of Windows almost completely...the steps necessary to get things going are much more logical and don't have some of Windows' weird restrictions...

Plus, you have to realise the difference in "attitude" involved...for instance, with X-Windows, your program needs to "connect to the X Server" ...something not done in Windows...but, then, the X-Windows solution is a whole lot better...what makes something an "X-Windows application" is the fact that it connects to an X server...under Windows, it's because the "GUI" bit is enabled in the headers...Windows is "hard-wired" to do local applications running on a single machine and you have to "jump through hoops" for any "remote" operations...while X-Windows is "client / server" from the ground up...so, yes, you have to "connect to an X server" but, then again, this also means you can connect to an X server on the other side of the planet over a broadband link...it means that you can connect to more than one server at the same time (one program controlling the output on multiple machines :)...and so on and so forth..."remote" applications simply require connecting to a "remote" server...and then that's it...otherwise, the rest of the application code is identical...you do not need to make any "allowances" because X-Windows does those for you (when it sees that the server is "remote", it automatically handles passing the stuff over the connection between client and server :)..

Also, there's other points of interest...in Windows, you either have a "console" or "GUI" application...if you choose "GUI" then even if it's started from a command prompt, the process doesn't inherit the "standard handles" (this is a very annoying thing I've encountered, trying to create "dual mode" programs...programs that act as command-line utilities if given command-lines but when given nothing on the command-line, they instead open up a GUI interface...so that you can use it "batch" or "interactive" as is most appropriate :)...again, X-Windows isn't really "thinking" like this...ordinary command-line applications and X-Windows applications are not any different...the difference comes about exclusively because you make a call to "open a connection to an X server" (or, as noted, to more than one X server for some really mad distributed functionality :)...you link your programs with XLib simply to get the function that opens the connection (if you knew the exact details of the "X protocol" then you can theoretically by-pass this and do it yourself...but that's "non-portable" – which is more of a concern on X-Windows because it really does run on PCs, Sparcs, Silicon

Graphics workstations and so forth :)...

Otherwise, there's no "window class" nonsense...nor is there "window procedure" nonsense...once connected to the X server, you need a "message queue" (but that's somewhat non-optional, if you think about it, without a "message queue" how exactly are you expecting to receive messages? ;) and then you just pop those messages off and react to them...when you want a window, you simply "CreateWindow" without any nonsense "window class" beforehand...that's how it should work...

If you want "window procedures" then simply create an ordinary procedure that accepts information about a message, containing one of those big "switch" statements...and then jump to that procedure with the new messages...on the other hand, you can deal with the messages all inside "main", if you like...or perhaps have "common" procedures between different types of windows...which is, of course, how you'd also get any "window class" functionality...if you want a bunch of windows to all be of the same "class" then, sure, write a "window procedure" and then pass those same messages to the same procedure (which'll react in the same way to those messages, making them all of the "same class" :)...

On the other hand, you can structure them completely differently, if you like...this isn't "overhead" and "restriction" in the same way as Windows' "window class" and "window procedure" nonsense is...this is simply choice because X-Windows provides the basic services and then it's up to the program to format these as is most appropriate to the task in hand...

Plus, X-Windows has a more sensible attitude to "messages" ...under Windows, every application gets every message...even if your program only processes WM\_CREATE, WM\_PAINT and WM\_DESTROY, it still gets a stream of WM\_MOUSEMOVE, WM\_SETCURSOR and so forth...what does it do with these other messages? Why, it wastes time simply sending them "return to sender" with "DefWindowProc" ...if you stop and think about this, then a simple easing up on the "control freak" attitude produces a far more sensible approach...the application knows what messages it processes, right? If you send it anything else, then it's just "return to sender" that it was slightly pointless sending it in the first place...

Well, though X-Windows isn't perfect by any means (only includes / excludes "ranges" of messages rather than individual ones), it at least allows an application to simply say "look, I don't ever read the mouse so there's totally no point sending mouse messages to this window"...of course, X-Windows has the right attitude here because all these messages may be being sent over a 56K modem connection to the other side of the planet...it can't really afford to be sending pointless messages for no good reason...hence, there is a means to simply specify "don't need those kinds of messages, don't even bother sending them"...

## alt.lang.asm: Re: OT: my new PC rocks!!

If someone out there is considering making their own GUI system, then I'd strongly advise that you take X-Windows attitude as a "role model" here rather than Windows...in fact, take it to the extreme once more, I'd suggest...default to sending `_NO_` messages (except, perhaps, "obvious" ones like "window creation", "window destroy", etc. ;) `_UNLESS_` the application specifically requests it...and the perfect opportunity there is to have the application simply specify what messages it actually processes while creating a message queue...but, also, include some "AcceptMessages" API where you can turn messages – individual or "ranges", by having a "start" and "end" parameter which can simply be the same message value for individual messages – "on" and "off"...there's the possibility that a window `_may_` initially check the mouse location once or twice and then never bothers ever again and so forth...

Windows spends an awful lot of time sending pointless messages...this change of style alone will win any GUI masses of extra processing time...there's also the possibility to "propagate" this benefit downwards...that is, a window doesn't accept mouse messages...well, not only does the GUI not need to bother that window with mouse messages but the OS can even inform the mouse driver not to bother the OS when it's in a certain area of the screen...the "pointless messages" can be propagated backwards to `_stop_` as soon as possible...apply this throughout the system from the ground up – plus, Windows wastes thousands of cycles in simple "overhead" handling a message, which you could work to improve upon – and you can "release" so much extra CPU power into a system that's far from insignificant...ever seen the red "kernel time" on Windows' task manager's graph of where the CPU power is going? Ideally, if you were to have a similar graph in some OS you wrote, then the thing to aim for is that the red graph is hardly there at all...it would be impossible to make it completely vanish but that, in a sense, is the "ideal" you should be working towards...an OS is `_BAD_` and `_WRONG_` when it's using CPU time...necessarily, it does have to use some in order to keep the system going...but the system is the `_applications_` – the OS merely there to "accomodate" them – and they should effectively totally rule the system as far as is possible...that understanding and attitude is `_severely missing_` from Microsoft and is one place where some other GUI / OS can `_do things properly_` so as to begin to blow Windows out of the water...because, yes, that's the thing that is also missing...people see Goliath and truly believe he's undefeatable...this is something that I don't believe at all...now, just as with the fabled Goliath, you don't stand a chance in a straight-out battle (the soldiers before David attempted that and failed miserably ;)...but, whether the "Art of War" actually says it like this or not, the best warrior wins without fighting (or by doing the least amount of fighting necessary)...

David, of course, was `_guaranteed_` to win...now, that may initially sound odd but if you look at the story, it was actually `_inevitable_`...this is the underlying message in the story that's

Re: OT: my new PC rocks!!

worth making obvious and clear...the reason being that the soldiers before David walked up to Goliath – within Goliath's reach – and tried to simply "out-strength" a giant when, clearly, he has more strength than ten of these soldiers put together...but note that David's approach was different and that was why he defeated Goliath...he used a sling...a "missile weapon" that could cause damage from afar within ever putting David in danger...just a case of David slinging stones until one of them hits that "right spot" that brings the giant down...many people understand the "even the smallest can defeat the biggest" part of this tale but many overlook the actual reason why David was sure to win...though his attacks may not be as strong as Goliath's, Goliath was unable to strike so all that strength he had amounted to absolutely nothing because he could never use it...however "small and weak" David's attacks in comparison, he was able to make his strikes...eventually, he would find that "weak spot" which takes him down in one blow...or simply pelt so many stones at Goliath that Goliath gives up or gets slowly and surely stoned to death...

Feel free to be impressed by Goliath's size but, if he's "a bit thick" – all brawn and no brains – then fear is completely inappropriate...and – \*ahem\* – "640KB is enough for anyone"? Need we say any more? ;)

> *I haven't done anything*  
> *with Xwindows yet – still haven't got "console mode" figured out...*

It's not particularly difficult...especially if you've gotten the rough idea of what all that "event-driven" nonsense is from writing some Windows code...in essence, it's just a program that waits for messages and reacts to them as they come...

And, in fact, in X-Windows' case, it isn't as weird as Windows in insisting that everything is in "window procedures" and so forth that it's easier to make the transition with X-Windows...other than linking to the XLib library (just so that you're linking to the API functions :), then it's no different – except for the functions it calls and the "wait for messages" structure – to other Linux applications...

> > *...and doing it all on Linux*  
> > *would be a pain because all my best development tools are on the*  
> > *Windows machine...*  
>  
> *Curiously, the Unix-folk claim that \*their\* tools are better – and*  
*have*  
> *invented Cygwin so they won't have to leave 'em behind when*  
*developing*  
> *for Windows! The famously user-friendly Unix interface bolted onto*  
*the*  
> *lean, mean, stable Windows chassis. A dream environment! (a*

nightmare's  
> *a dream, innit?*)

Oh, don't get me wrong...UNIX tools are better in some regards...that is, "GREP" is one of those indispensable UNIX tools...they are able to do some pretty impressive stuff from a simple command-line on occasion...and there's just tons of these "short and sweet" utilities (having "many small" which you jigsaw together to do your task, is usually far more useful and flexible than "one big" – which tends to be the Windows way – in general :).)

> *What would you miss most that you can't have in Linux?*

My MP3 collection, my rather large "tech-docs" folder where I store up all my references (makes no impact to my many tens of GB hard disk on the Windows machine, would be rather large though for the smaller "second machine"), Windows Solitaire and Spider Solitaire games, the text editor that I've gotten so used to using that I can do things really quickly with it, my "bells and whistles" graphic editor program (professional quality and wasn't cheap to get ;), etc....

Yes, nothing technical...that's the grand mistake "geeks" always make..."technical" and "techno-fix" is often the least important thing...Windows and the stuff I have on the machine are "comfortable" and I'm used to them...and, therefore, I can develop happily and quickly...

Oh, trust me...I've used Cygwin to run Linux GCC on Windows in order to cross-develop for a PDA, for example...I've cross-developed onto circuit boards that have no "monitor" or anything (switching on and off small LEDs on the front panel to let me know things were working properly ;)...the other day, I was bored round a friend's house and they had a PC but absolutely NO development tools at all and I decided to waste time – had to wait around for hours one day – waiting by challenging myself to a simple but tricky task: "using a command prompt and debug, slowly build a hex tool, then an assembler then work up to writing some simple DOS-based .COM game" and the trick being that I had NO references, NO tools or anything...only what comes "out of the box" – the command prompt commands, DEBUG, etc. – with any Windows installation (never actually completed this task because it takes longer to do than I actually had to wait...but it was a way to waste time while I was waiting...and I had the plan in my head that it was completely possible...it just would probably take days to complete on the first attempt...using DEBUG itself in order to get the opcodes for the assembler...only those DOS / BIOS interrupts – like "mov ah, 00h; int 16h" for keyboard input – that I could actually remember off the top of my head :)...also, I started out programming on those little 8-biters where 64KB was "elephant memory"...

I can work, if necessary, in some bloody awful "conditions"...but, having two machines – one running Windows (which I've built up into a

nice, comfortable system :) and one running Linux – is the perfect solution...note that when Linux is the better choice, I simply use the other machine for that task...but, well, I use Windows primarily and it's my "main machine"...plus, there's the "little things"...like, say, Randy's PDF documentation...reading PDFs is easy in Windows, I've got the plug-ins already but it would require a lot of downloading and so forth to get a Linux installation up to the same levels of "convenience"...IF I only had Linux then that's exactly what I'd do...and, over time, if I add to the Linux system then I could perhaps get it up to a level where it doesn't particularly matter which I use...and cross-developing, I AM using Linux and an "X-term" completely for the actual compiling and testing and stuff...but, well, though "gedit" is more or less "Notepad which can handle multiple files at the same time", it's just not as convenient or am I comfortable using it as the IDE that I've had for years that I know it backwards...

This is the really weird thing in the Linux world...they are constantly on the defensive and so forth...you know, if someone isn't singing their praises that Linux is "perfect" in ALL possible situations, then I'm "evil"...I like the bloody thing...that's why I am – if people are actually remembering what this is all about – developing X-Windows applications for Linux...how I'm Hoping to help Randy get X-Windows properly into HLA that doing GUI applications is exactly as "easy" and "without modification" that HLA can already do for "console" applications...that, in fact, is exactly an exercise in leveling the field between the two so that, yup, it doesn't matter which you use...

I like that KDE thing (the best yet is that you can get it to put a menubar over the top of the screen like an Apple Mac...I've always liked that style – certainly is more welcoming and useful to "newbies" to be able to read the available commands off a menu – and KDE is "politically correct" enough in its options that you can get it to act like any GUI...and I was using plenty of GUIs before Windows, that I'm NOT always impressed with the "Windows way" and DO KNOW that there are better ways...Windows' "taskbar" and "resizing borders" is the better on that score, Apple's "menu at the top" is better on other scores...and I totally prefer a desktop where the drives are listed as icons rather than some strange abstract "My Computer" (which is a slightly presumptuous name, I've always thought...what if it's a machine in a school or business or at a friend's house? Then it's NOT "My Computer" at all...although, it can be excused because you know what it's trying to say – being "colloquial" in the name deliberately to make it sound "friendlier" – it's a bit presumptuous and can, therefore, often be simply factually incorrect ;)...

This ain't in any way some "anti-Linux" attack...far from it...just that when things aren't right, I always speak my mind...things aren't 100% right in all departments...there will NEVER ever be such a thing in existence on this Earth that fits that description, as

"NOTHING is perfect"...I think the "issue" with Linux is that people actually take comments about Linux personally...you know, I say "oh, I can't stand that 'vi' editor" and it's like I've just called their mother "ugly" or something...look, I can't stand it...doesn't mean that other people can't like it and think it's the best thing since sliced bread...it's just the way things are...no need to take it as a personal assault on all of creation or anything...chill out, Linux people...if anything, I totally "come in peace" here to help out...this whole "Spanish Inquisition" is part of all the "voodoo" nonsense that I was actually pointing out...you just don't expect such things when you're just trying to make honest comments and ask simple questions and make some jokes to Hopefully bring a smile to someone's face somewhere :)

> > *I'm also wondering whether to write a book on the psychology of how people describe their machines...*  
>  
> *Mac users get to say, "Mine's blue!" Well, I suppose we could spray-paint 'em...*

Ooh, those iMacs are works of art...actually, literally, because they won design awards that covered a whole bunch of stuff that it really did have stiff competition that winning with something so usually "not even considered" as the design of a computer is earned and deserved...I even saw a bath in a magazine which actually was "inspired" by the iMac design, having that same "semi-transparent blue" plastic bit thingy...so you could colour co-ordinate yourself while relaxing in the bath using your iMac (though, hmmm, encouraging the use of electronic equipment in the bathroom? Someone's not being totally "safety conscious" there, eh? ;)...much like the Playstation's joypad - which also won awards for brilliant design - I've had to agree with the judge's decisions...these two things really were fantastic design achievements...really cool-looking but still perfectly practical and functional...

Mind you, have you seen the "explosion" of PCs with moulded plastic blue bits and odd curved shapes, trying to copy the iMac? Slightly pathetic, really, as none of them come close...they are only thinking "jump on the bandwagon", they aren't thinking "art" (but also "functional"...because the hard drive and stuff is actually also inside that monitor, meaning that the entire system is nice and easy to carry around too...as it's all in one place and only needs the keyboard and mouse attached...though, with wireless stuff, even that won't be necessary ;) which is what made the original design so cool...

I'm not quite so impressed with the "lampstand" design...it doesn't quite look as "cool-looking" as the original iMac designs (which, by the way, you can get in a silly amount of different colours...there's even "zebra stripes" and "dalmation spots" and mad-looking ones like

alt.lang.asm: Re: OT: my new PC rocks!!

that :)...but it's still innovative and impressive design...just the original iMac designs were so brilliant, even something great looks a bit crap in comparison to it...

When I went into the store to buy my current PC, there were all these "bad iMac impersonation" PCs...you know, strange curved bits of coloured plastic for no particular reason but to try to pretend that the PC was actually an iMac...and when I ordered my PC, he asked what sort of box I wanted it put into (because it was one of those "put together to order" things :) and I looked at all the boxes...and the way I put it was: "\_IF\_ there's one here that looks as cool as an iMac then I'll take it...but if it's not quite that good, then go to the total other extreme and give me a plain boring beige-coloured box"...you know, it's either drop dead gorgeous looking or, if that can't be managed, then just don't bother with "looks" at all and give me something boring, plain and simple...in short, I said "I either want the real deal or don't bother"...the box I have, by the way, is plain with only a few small bits of blue plastic on it (which are all removable :)...I'm pleased to say that the guy behind the counter \_totally\_ understood what I meant and made the right choice...you know, either you drive a drop dead gorgeous Porsche / Ferrari or you drive a "functional but ugly" Skoda or whatever...the real deal or nothing at all...so, to sum it up in one sentence: "I have no time for Toyota" ;)

Anyway, what's wrong with beige? I mean, it's the perfect "camouflage" colour for hiding vomit stains...very "functional"...hehehe ;)

Beth :)