

GÖMÜLÜ SİSTEMLER VE LINUX

Hazırlayan

Murat Demirten <murat@debian.org>

2. Linux ve Özgür Yazılım Şenliği, 3 Mayıs 2003

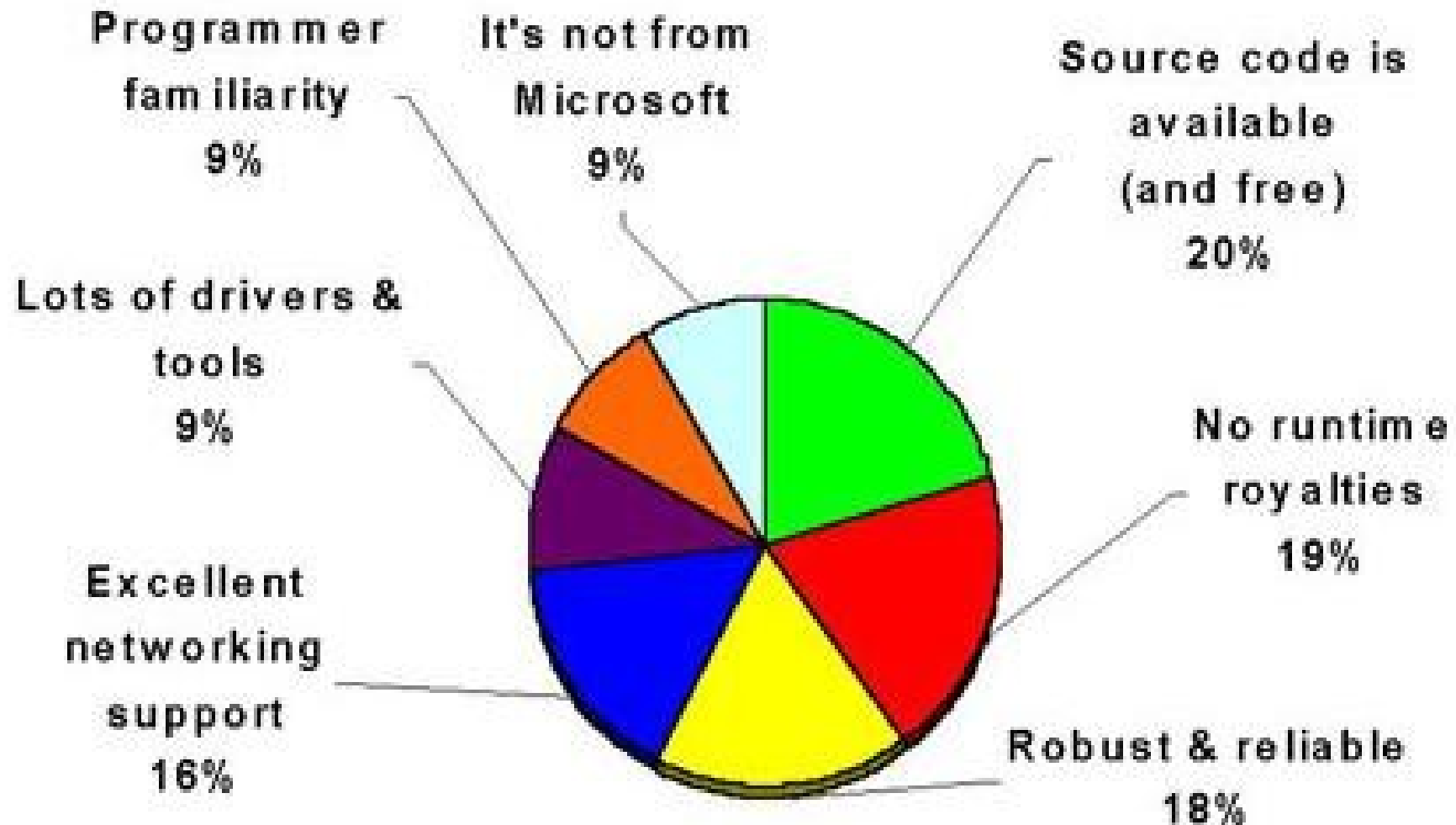
Gömülü (Embedded) Linux?

- Gömülü (Embedded) Linux Sistemler kavramı henüz bir kaç yıllık geçmişe sahip.
- Gömülü sistemler için yazılım/donanım ürünleri geliştiren firmalar, bu alandaki gelişmeler karşısında stratejilerini yeniden düzenliyor veya tamamen değiştiriyorlar.
- Linux, bu pazarda kullanımı en hızlı ve istikrârlı artan işletim sistemi oldu.

Neden Gömülü Linux?

- Güç, verimlilik, kararlılık
- Geniş işlemci ailesi desteği
- Geniş donanım desteği
- Ağ iletişimi için gelişmiş özellikler
- Açık kaynak kodu
- Hızlı gelişen, binlerce kaynak kodu açık uygulamanın çalıştığı bir platform

Neden Linux – Anket



Linux'un Alternatifleri ve Problemleri

- Eski dost DOS: Günümüz teknoloji ve işlevsellik gereklerini karşılamaktan çok uzak
- Ev yapımı kernel: Teknolojinin takibi çok zor ve geliştirim maliyeti fazla
- Windows: Verimlilik problemi, büyük, hantal, pahalı
- UNIX: Embeddable değil, pahalı
- Özel RTOS'lar: Pahalı, standartlarla uyumsuzluk

Başarılı Örnekler



PDA



Gateway, server, wireless a.p.



Diğer

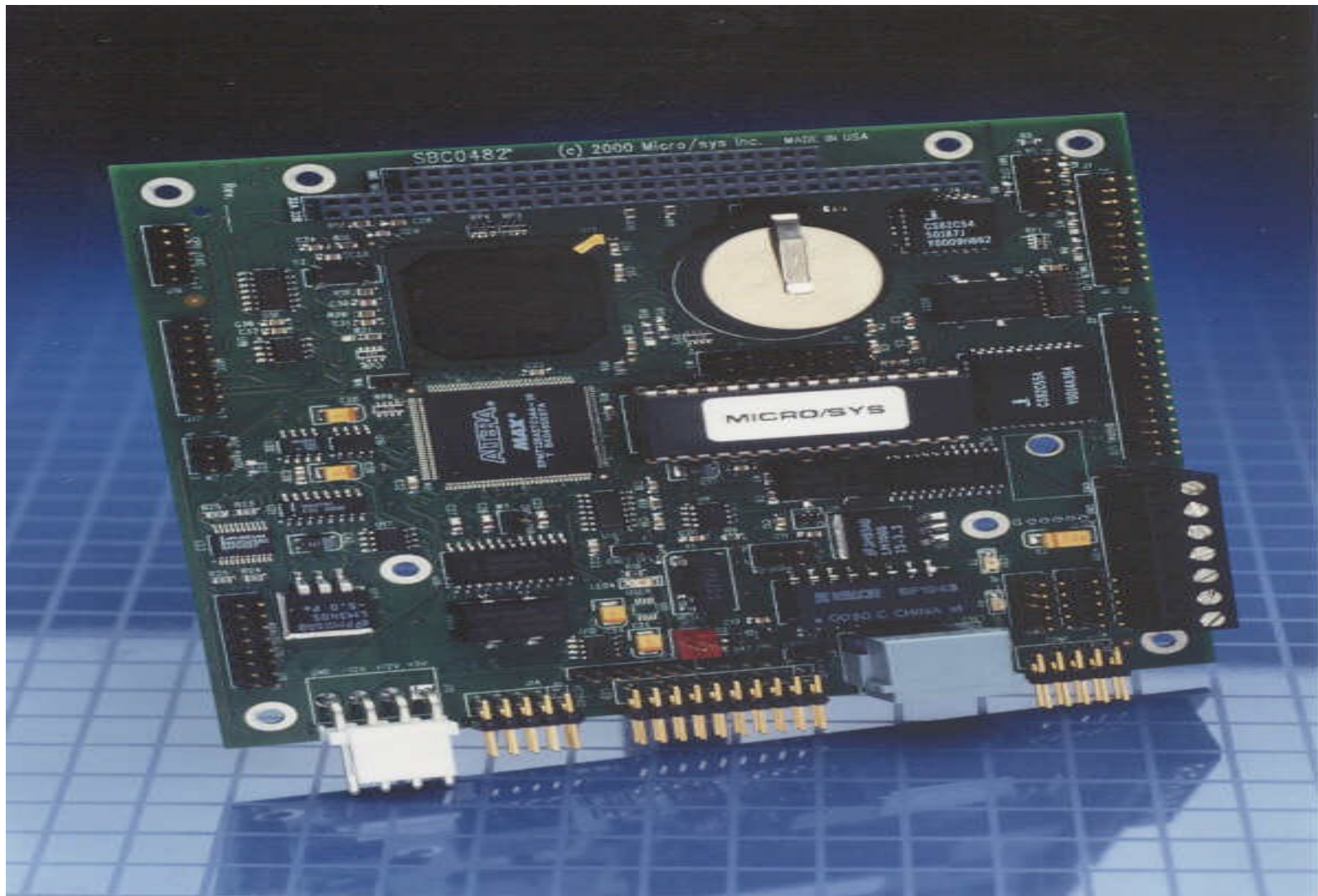


Webpad

Gömülü Linux Dağıtımını Geliştirme

- Kullanılacak Donanım: Single Board Computer, PC 104, PC 104+ vb.
- Tasarım prensipleri: DiskOnModule, DiskOnChip kısıtları
- Geliştirilecek sistemden beklentilerin netleştirilmesi ve tasarımının yapılması
- Geliştirme ortamı: Linux yüklü bilgisayar (internet bağlantısı şart :)

Single Board Computer (SBC)



Gerekli Bağlantılar

- SBC için gerekli güç bağlantısının yapılması
- İki sistem arasında dosya taşımak için disketler
- SBC için disket sürücü bağlantısı, CDROM takılabiliyorsa gerekli bağlantıların hazırlanması
- İstendiği takdirde seri porttan dosya transferi için null modem kablosu
- Ethernet desteği var ise cross veya normal ağ UTP kablosu

Sistemi Boot Etme

- Öncelikle SBC'yi disketten veya cdrom üzerinden Linux ile boot etmeliyiz.
- Cd-rom desteğinin çoğu durumda olmayacağını düşünürsek, sistemi boot edebilmek için herhangi bir Linux dağıtımının boot ve root disketlerini oluşturup sistemi bu disketler ile açmalıyız.

Disk Bölümleme

- Herhangi bir yöntemle SBC Linux ile boot edildikten sonra konsol ekranına geçilir.
- Fdisk ile hda veya hdc üzerine bir Linux disk bölümü yaratılır (Tip: 83).
- Swap bölümü kesinlikle oluşturulmaz. Kullanılan disk teknolojileri nedeniyle swap bölümü, diskinizin (DiskOnModule, DiskOnChip) ömrünü azaltmakla kalmaz, bitirir. Wear-levelling gibi teknolojiler geliştirilse de henüz NOR Flash bellekler için 100.000 write ömrü aşılamadı.

Dosya Sistemi Oluşturma

- Kullanılacak sisteme uygun dosya sistemi seçimi yapılmalıdır.
- Minix dosya sistemi ext2'ye göre daha hızlıdır, sisteme daha az yük getirir ve hemen her işlem için yeterlidir. Ancak bir minix bölümünün büyüklüğü maksimum 64 MB olabilir, dosya adları maksimum 31 karakterdir vb.
- Ext2 dosya sistemi için,
`mkfs.ext2 /dev/hda1 -b 1024 -m 1`

İlk Mount, Dizinlerin Oluşturulması

- Yeni bir dizin yaratılıp disk mount edilir
`mkdir /EMBEDDED`
`mount /dev/hda1 /EMBEDDED`
- Artık hedef sistem /EMBEDDED altına mount edilmiş durumdadır. Tüm geliştirme bu dizin altında yapılır. Bu dizine geçildikten sonra temel dizinler oluşturulur:
`mkdir -p bin boot dev etc floppy home`
`root lib proc sbin tmp usr var usr/bin`
`usr/lib usr/sbin`

Kütüphanelerin Taşınması

- Linux PC'den `libc.so.6` `ld-linux.so.2` gibi temel kütüphaneler `lib/` dizini altına disket ile taşınmalıdır.
- Bu iki kütüphane dosyasına çalıştırma hakları verilmiş olmalıdır:
`chmod 755 lib/libc.so.6`
`chmod 755 lib/ld-linux.so.2`

Busybox ve Tinylogin Kullanımı

- Busybox ve Tinylogin: Sık kullanılan uygulamaların, temel özellikleriyle yeniden yazılarak disk alanından kazanç sağlanması için oluşturulmuştur.
- www.busybox.net adresinden her iki program indirilip, istenilen özelliklerle derlenir. Bu iki program toplamda 300-400 Kb arası yer kaplar, ancak 4-8 Mb kadar kazanç sağlar.
- Linkleri ile birlikte her iki program /EMBEDDED altına kopyalanır.

Aygıt Dosyalarının Oluşturulması

- Sistemdeki /dev dizini altında aygıt dosyalarının oluşturulması gerekir. Bunun için çalışan bir Linux sistemdeki MAKEDEV betiği kopyalanıp aygıtlar oluşturulabilir, sistemin boot edildiği disketlerin oluşturmuş olduğu /dev dizini kopyalanabilir veya major, minor numaraları verilerek tek tek gerekli tüm aygıt dosyaları yaratılabilir.

/etc Altındaki Ayarlamalar

- Boot işleminde kernel işini bitirdikten sonra `/sbin/init` programını çalıştırır. Program `/etc/inittab` ayar dosyasını okur.
- Busybox ile gelen `init`'in `inittab` formatı biraz değişiktir (Busybox'ta `runlevel` kavramı yoktur)
- Bu dosyada aşağıdaki gibi kayıtlar olmalıdır:
`::sysinit:/etc/init.d/rcS`
`tty1::askfirst:/bin/sh`
`tty2::respawn:/sbin/getty 38400 tty2`

/etc Altındaki Ayarlamalar...

- `chroot /EMBEDDED /bin/sh` ile kök dizin / yeni oluşturulan sisteme kaydırılır.
- Burada temel gurupların yer aldığı `/etc/group` dosyası oluşturulur. `/etc/passwd` dosyasına root için bir kayıt girilir. Ardından `adduser` ile sisteme kullanıcı eklenebilir, root şifresi değiştirilebilir.
- `ld.so.conf`, `fstab`, `nsswitch.conf`, `hosts`, `profile` dosyaları oluşturulur.

/etc Altındaki Ayarlamalar...

- Inittab dosyasında sysinit satırıyla belirtilen `/etc/init.d/rcS` dosyası boot esnasında çalıştırılacak ilk programdır. Bu açılış programının oluşturulması gereklidir.
- rcS programı root bölümü read-write modunda mount etmeli, proc pseudo dosya sistemini `/proc` dizini altına ayarlamalı, ağ ayarlarını yapmalı ve sistemin açılışında olması gereken tüm işlemler için ilgili programları tetiklemeli, çalışması bittiğinde sistemi tamamen hazır edebilmelidir.

Sistemi Boot Edebilir Hale Getirme

- Sistemin boot edebilmesi için bir boot yükleyiciye ihtiyaç vardır. Eğer lilo kullanılacaksa busybox içerisinde çıkmadığı için bir Linux sistemden kopyalanmalıdır.
- Linux sistemden program taşımada izlenecek yol, öncelikle ldd ile programın kullandığı kütüphanelerin listesini almak, sonra SBC üzerindeki sisteme bunlardan eksik olan kütüphaneleri ve programın kendisini kopyalarak programı çalışır hale getirmek olmalıdır.

Sistemi Boot Edebilir Hale Getirme...

- Lilo gibi bazı programların çalışması için ek dosyaların kopyalanması gerekebilir. Örneğin lilo için `/boot/boot-text.b` vb.
- SBC sisteminde kullanılacak çekirdek, gerekli özellikleri içerecek şekilde bir Linux sistemde önceden derlenmeli ve hazırlanmalıdır.
- Hazırlanan çekirdek `/boot` dizini altına taşındıktan sonra uygun bir `/etc/lilo.conf` dosyası hazırlanmalıdır. Ardından lilo komutu ile MBR yeniden oluşturulmalıdır.

Dosya Transferleri

- Sistem ilk geliştirilene kadar sürekli dosya transferi yapmak zorunda kalacaksınız. Bir defalık bu eziyeti, biraz olsun kolaylaştırmak amacıyla öncelikle minicom kurup seri porttan veya ftpd kurup ağ üzerinden dosya gönderebilmek için gerekli programları ayarlamalısınız.
- Ftpd gibi programların PAM desteğini inaktifleştirerek derlemeniz ve sisteme kurmanız önerilir. Diğer durumda PAM için de bir miktar ver kaybetmek zorunda kalırsınız.

Program Kurulumları

- Sisteme Xwindow vb. Programlarının kurulumu için bir Linux sistemi örnek alabilirsiniz. Bu sistemdeki dosyaları sırasıyla kopyalayarak SBC üzerinde çalışır bir sistem oluşturmanız mümkündür.
- SBC üzerine atılan programlar mutlaka “strip” edilmelidir. Böylelikle gereksiz semboller obje dosyaların içerisinde yer almayacak, daha az yer kaplayacaktır.

Read-Only & Read-Write

- Disk ömrünü artırmak amacıyla, eğer mümkünse sistemi sürekli read-only modda çalışacak şekilde ayarlayınız.
- Ramdisk kullanımı ile belleğinizin bir bölümünü disk gibi kullanabilir, gerektiğinde sistemi read-write moda geçirebilirsiniz.
- Tasarımı bu şekilde yaptığınızda çift aşamalı açılış için initrd veya devfs sistemlerini kullanmak gerekecektir.

Teşekkürler

Elbette konu çok daha uzun, ancak zamanımız kısıtlı :(

Sorularınız için murat@debian.org adresine yazabilirsiniz.