T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

UYGULAMA RAPORUNUN BAŞLIĞI

Taha Baran AKTİ

UYGULAMA RAPORU

Bilgisayar Mühendisliği Anabilim Dalı
Tezsiz Yüksek Lisans Programı

01-2025
KONYA

UYGULAMA RAPORU KABUL VE ONAYI

Taha Baran AKTİ tarafından hazırlanan “Minimalist Bir Linux Dağıtımı Geliştirilmesi” adlı uygulama raporu …/…/… tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda TEZSİZ YÜKSEK LİSANS UYGULAMA RAPORU olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Danışman Prof. Dr. Mustafa Servet KIRAN	……..
Üye Prof. Dr. Erkan ÜLKER	……..

UYGULAMA RAPORU BİLDİRİMİ

Bu uygulama raporundaki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve uygulama raporu yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by application report rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Taha Baran AKTİ

Tarih:

ÖZET

TEZSİZ YÜKSEK LİSANS UYGULAMA RAPORU

MİNİMALİST BİR LİNUX DAĞITIMI GELİŞTİRİLMESİ

Taha Baran AKTİ

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Erkan ÜLKER

2024, 48 Sayfa

Jüri
Prof. Dr. Erkan ÜLKER
Prof. Dr. Mustafa Servet KIRAN

Linux, açık kaynaklı bir işletim sistemi olması, düşük donanım gereksinimlerinde bile yüksek performans sunması ve güvenilirliği ile öne çıkmaktadır. Ancak pek çok Linux dağıtımı, gereksiz program ve işlevler içermesi nedeniyle performans kaybı ve fazla disk alanı kullanımına yol açabilmektedir.

Bu çalışmada, sıfırdan bir Linux çekirdeği kullanılarak, yalnızca gerekli bileşenleri içeren, hafif ve sunucu odaklı bir Linux dağıtımı geliştirilmiştir. Çalışma boyunca izlenen yöntemler ve elde edilen sonuçlar detaylı şekilde sunulmuştur.

Anahtar Kelimeler: Açık Kaynak, Çekirdek, Linux, İşletim Sistemi, Kabuk, Sanallaştırma

ABSTRACT

NON-THESIS MASTER'S DEGREE APPLICATION REPORT

DEVELOPMENT OF A MINIMALIST LINUX DISTRO

Taha Baran AKTİ

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Advisor: Prof. Dr. Erkan ÜLKER

2024, 48 Pages

Jury
Prof. Dr. Erkan ÜLKER
Prof Dr. Mustafa Servet KIRAN

Linux stands out as an open-source operating system, offering high performance even on low-spec hardware and ensuring reliability. However, many Linux distributions include unnecessary programs and functions, which can lead to performance loss and excessive disk space usage.

In this study, a lightweight, server-oriented Linux distribution was developed from scratch using the Linux kernel and including only the essential components. The methodology followed throughout the study and the obtained results are presented in detail.

Keywords: Linux, Kernel, Open Source, Operation System, Shell, Virtualization

ÖNSÖZ

Bu çalışmayı yürütürken bana destek olan danışman hocama ve Yüksek Lisans süreci boyunca ders aldığım hocalarıma teşekkür ederim. Bu proje, hem akademik hedeflerimi gerçekleştirmem hemde açık kaynaklı sistemlerin gücünü göstermek açısından benim için önemli bir adım olmuştur. Sürecin her aşamasında edindiğim deneyimler, bana yalnızca teknik bilgiler kazandırmakla kalmayıp, aynı zamanda problem çözme yeteneğimi de geliştirmiştir.

Bu tezde sunulan bilgilerin, açık kaynak dünyasına katkıda bulunmasını ve gelecekteki çalışmalara ışık tutmasını dilerim.

Taha Baran AKTİ
KONYA-2025

İÇİNDEKİLER

ŞEKİLLER

KISALTMALAR

- LFS: Linux From Scratch, sıfırdan Linux sistemi oluşturma rehberi.

- GRUB: Grand Unified Bootloader, işletim sistemi yükleyicisi.

- KERNEL: Çekirdek, işletim sisteminin merkezi bileşeni.

- FHS: Filesystem Hierarchy Standard, Linux dosya sistemi düzeni için standartlar.

- BASH: Bourne Again Shell, Linux komut satırı kabuğu.

- APT: Advanced Package Tool, Debian tabanlı sistemlerde paket yönetim aracı.

- SSH: Secure Shell, güvenli uzaktan erişim protokolü.

- OS: Operating System (İşletim Sistemi).

- CLI: Command Line Interface (Komut Satırı Arayüzü).

- GUI: Graphical User Interface (Grafiksel Kullanıcı Arayüzü).

- KB: Kilobyte.

- MB: Megabyte.

- GB: Gigabyte.

1. GİRİŞ

Bilgi teknolojilerinin hızlı gelişimi, işletim sistemlerinin daha fazla özelleştirilebilir, performans odaklı ve minimal hale gelmesi ihtiyacını doğurmuştur. Özellikle sınırlı donanım kaynaklarına sahip sistemlerde, yalnızca gerekli bileşenleri içeren minimalist tasarımlar büyük bir avantaj sağlamaktadır. Linux, açık kaynaklı ve modüler yapısıyla bu ihtiyaca yanıt veren esnek işletim sistemi seçeneklerinden biri olarak öne çıkmaktadır.

Minimalist Linux dağıtımları, sistem kaynaklarını optimize ederek düşük bellek ve işlemci kullanımı, hızlı başlangıç süreleri ve düşük disk alanı gereksinimi sunar. Ancak, mevcut dağıtımların çoğu genelleştirilmiş kullanım senaryolarına yönelik tasarlandığı için spesifik ihtiyaçlara yanıt verme konusunda sınırlı kalmaktadır. Bu çalışmanın amacı, yalnızca temel bileşenlere sahip, yüksek performanslı bir Linux dağıtımı geliştirme sürecini ele almaktır.

Bu proje kapsamında, gereksiz bileşenlerden arındırılmış bir Linux çekirdeği yapılandırılarak, minimum kaynak tüketimiyle maksimum verimlilik sağlayan bir sistem tasarlanmıştır. Çalışmada ayrıca çekirdek modifikasyonları, dosya sistemi yönetimi ve önyükleme süreçleri detaylı bir şekilde ele alınmıştır. Geliştirilen sistem, test sonuçlarıyla birlikte performans açısından analiz edilerek avantajları ve sınırlamaları ortaya konulmuştur.

Projenin devamında gereksiz kalabalık ve karışıklığı önlemek adına geliştirilecek dağıtıma “MyOs” adı ile hitap edilecektir.

1.1 MyOS'un Tanımı

MyOS, tamamen minimalist bir Linux tabanlı işletim sistemidir. Tasarımının merkezinde şu temel prensipler yer alır:

- **Minimal Bileşenler:** Sistemde yalnızca temel işlevler için gerekli olan bileşenler bulunur, gereksiz her şeyden arındırılmıştır.
- **Kaynak Optimizasyonu:** Bellek, CPU ve depolama alanını minimum seviyede tüketerek etkili bir performans sunar.
- **Sunucu Odaklı Tasarım:** Yüksek performans gerektiren sunucu ortamlarına uygun olarak tasarlanmıştır.

Bu prensipler, MyOS'u aşağıdaki senaryolar için özellikle uygun hale getirmektedir:

- **Sunucu Yönetimi:** Minimum kaynak tüketimiyle çok sayıda kullanıcıya hizmet sunabilecek bir sistem.
- **Test ve Geliştirme Ortamları:** Gömülü sistemlerde ya da yazılım testlerinde hafif ve esnek bir temel sağlar.
- **Eski Donanım Kullanımı:** Modern işletim sistemlerini desteklemeyen eski makineler için verimli bir çözüm sunar.

1.2 Projenin Amacı

Bu çalışmanın amacı, MyOS'un geliştirilme sürecini ve bu süreçte kullanılan teknik detayları detaylı bir şekilde ele almaktır. Çalışmanın hedefleri şunlardır:

- **Performans Artışı:**
    - Sistem açılış sürelerini 12 saniyenin altına düşürmek.
    - Minimum bellek ve CPU kullanımı ile maksimum verimlilik sağlamak.
- **Küçük Ayak İzi:**
    - Yalnızca gerekli bileşenleri barındırarak disk alanı kullanımını 500 MB altında tutmak.
- **Esneklik ve Özelleştirilebilirlik:**
    - Kullanıcıların, sistemlerini özelleştirme ihtiyacını karşılayan modüler ve seçenekler sunmak.

MyOS, geleneksel Linux dağıtımlarından farklı olarak tamamen minimalist tasarım ilkelerine dayanmaktadır. Bu tasarım anlayışı, teknik performans artışının yanı sıra ekonomik avantajlar da sağlar; donanım maliyetlerini düşürmek ve enerji verimliliği sağlamak gibi faydalar sunar.

1.3 Minimalist Linux Dağıtımları Üzerine Genel Bir Bakış

Minimalist Linux dağıtımları, sınırlı donanım kaynaklarına sahip sistemler için geliştirilmiştir. Bu dağıtımlar, yalnızca temel işlevlerin yer aldığı bir yapı sunarak kaynak kullanımını optimize eder ve kullanıcılara maksimum verimlilik sağlar.

Avantajlar:

- **Kaynak Verimliliği:** Daha düşük bellek, CPU ve disk alanı kullanımı.
- **Hızlı Başlangıç Süreleri:** Azaltılmış bileşenler sayesinde daha kısa yükleme süreleri.
- **Esneklik:** Kullanıcılar, sistemlerini kendi ihtiyaçlarına göre kolayca özelleştirilebilir.

Minimalist Linux Dağıtımlarından Örnekler:

1. **Alpine Linux:**
    - Konteyner ortamlarında yaygın olarak kullanılır.
    - Minimal boyut (5 MB) ve güvenlik odaklı tasarım.
2. **Arch Linux:**
    - Kullanıcı dostu özelleştirme seçenekleri sunar.
    - Minimal başlangıç kurulumuyla bilinir.
3. **Tiny Core Linux:**
    - Sadece 16 MB boyutundadır.
    - Eski donanımlarda dahi hızlı çalışabilir.

Bu dağıtımlar, MyOS'un geliştirilmesinde ilham kaynağı olmuş ancak MyOS, yalnızca gerekli bileşenlere odaklanması ile daha az kaynak tüketen bir alternatif sunmaktadır.

1.4 Tezin Yapısı

Bu tez, MyOS'un geliştirilme sürecini ve teknik analizini aşağıdaki başlıklar altında ele almaktadır:

- **Giriş:**
    - MyOS'un genel tanımı ve temel hedefleri.
    - Minimalist Linux dağıtımlarına dair genel bir bakış.
- **Literatür Taraması:**
    - Minimalist işletim sistemlerinin tarihsel gelişimi.

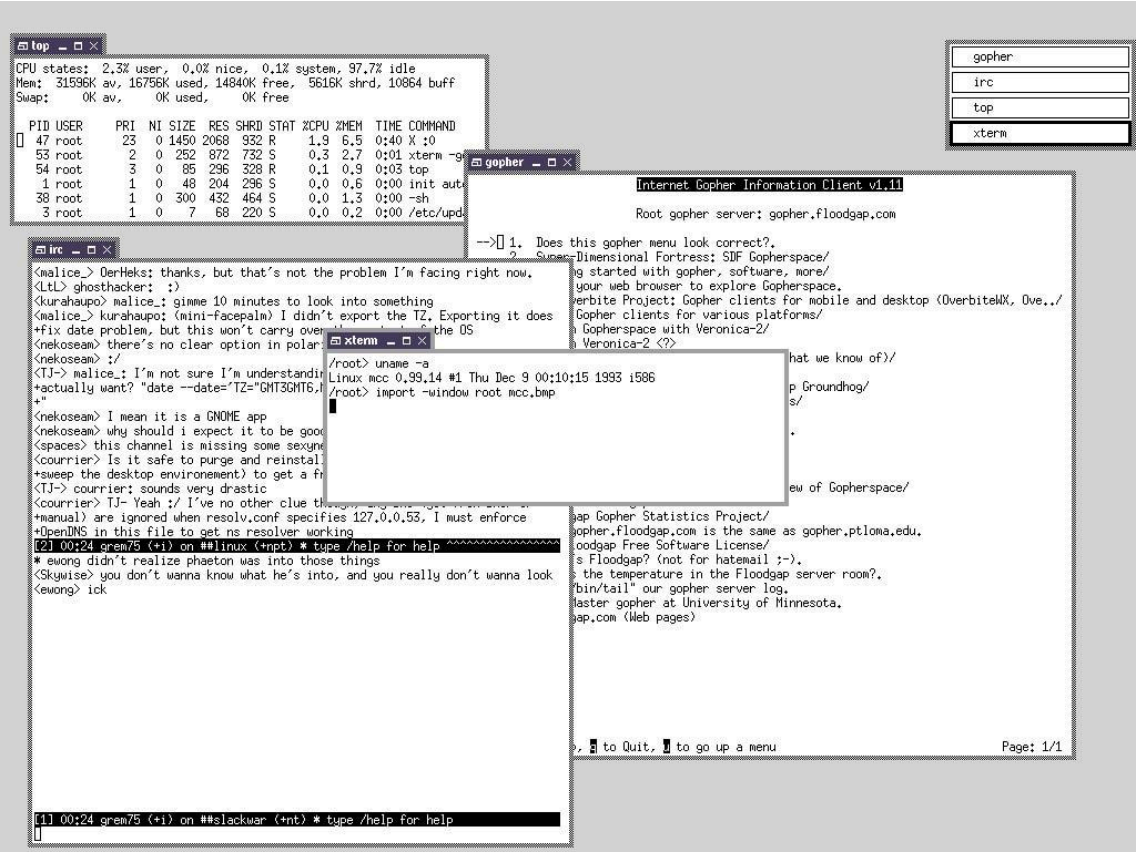
    - Teknik analiz ve benzer sistemlerle karşılaştırmalar.
- **Yöntem:**
    - MyOS geliştirilirken kullanılan araçlar, süreçler ve yaklaşımlar.
- **Kod Analizi:**
    - MyOS'un temel bileşenlerinin ve kod yapısının açıklamaları.
- **Sonuçlar ve Tartışma:**
    - Performans değerlendirmesi ve geliştirme sürecinde karşılaşılan zorluklar.
    - Gelecekteki geliştirme fırsatları ve öneriler.
- **Kaynakça ve Ekler:**
    - Kullanılan kaynaklar ve destekleyici bilgiler.

2. LİTERATÜR TARAMASI

2.1 Minimalist İşletim Sistemlerinin Tarihçesi

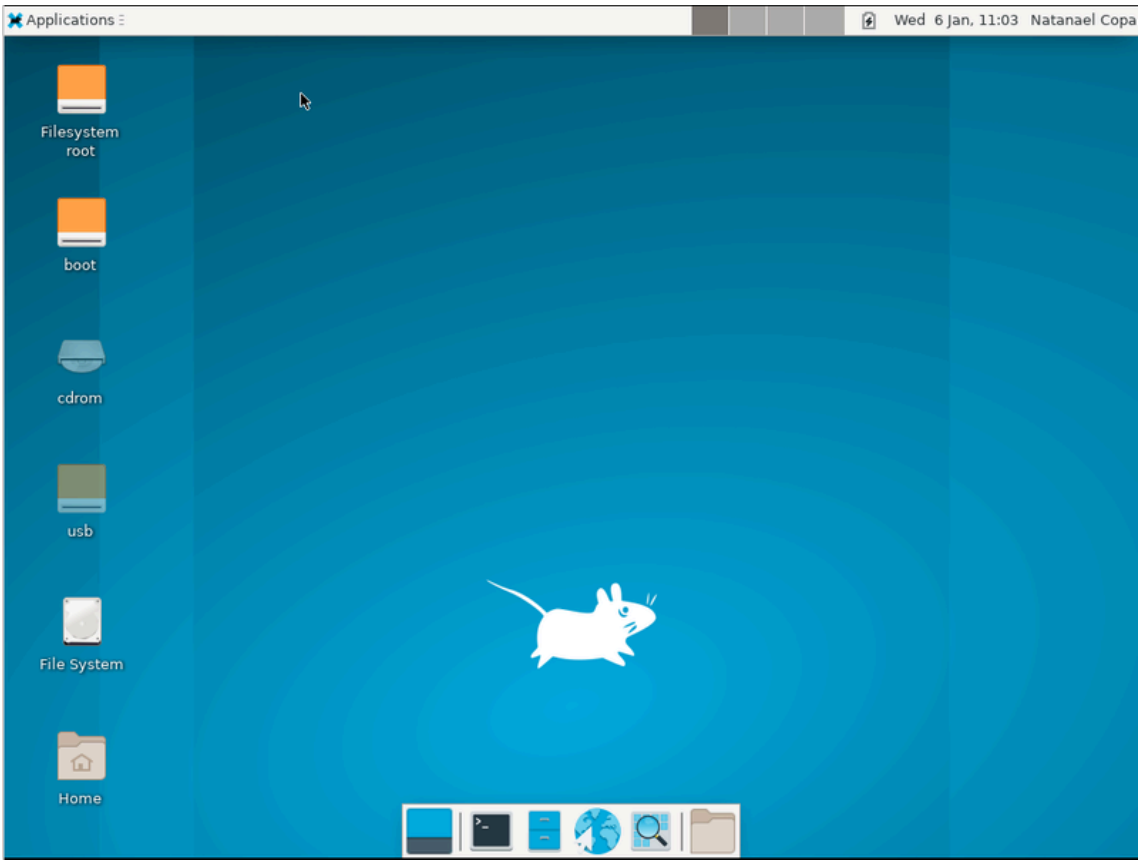
Minimalist işletim sistemleri, bilgisayarların ilk dönemlerinden itibaren önemli bir rol oynamıştır. Özellikle sınırlı donanım kaynaklarına sahip sistemlerde, yalnızca gerekli işlevleri barındıran minimalist tasarımlar tercih edilmiştir.

- **Erken Dönem Gelişmeler:**
    - **1970'ler ve UNIX:** Minimalist bir tasarım felsefesiyle geliştirilen UNIX, daha sonra birçok işletim sisteminin temelini oluşturmuştur.
    - **1980'ler ve Gömülü Sistemler:** Mikroişlemcilerin yaygınlaşmasıyla, gömülü sistemler için minimalist işletim sistemleri geliştirilmeye başlanmıştır.



Şekil 2.1 MCC Interim Linux 1992'de ortaya çıkan minimal bir dağıtım

- **Modern Minimalist İşletim Sistemleri:** Minimalist Linux dağıtımları, modern bilgi işlem ihtiyaçlarına uygun bir şekilde tasarlanmıştır. Öne çıkan örnekler:
    - **Alpine Linux:**
        - Küçük boyutuyla bilinir (5 MB).
        - Konteyner uygulamalarında yaygın olarak kullanılır.
    - **Tiny Core Linux:**
        - 16 MB'lık bir boyuta sahiptir ve eski donanımlarda bile çalışabilir.
    - **Arch Linux:**
        - Kullanıcı tarafından yapılandırılabilir bir temel sistem sunar.



Şekil 2.2 Xfce yüklenmiş Alpine Linux

Bu dağıtımlar, MyOS'un geliştirilmesi için temel referans noktaları olmuştur.

2.2 Linux Çekirdeği ve Modüler Yapı

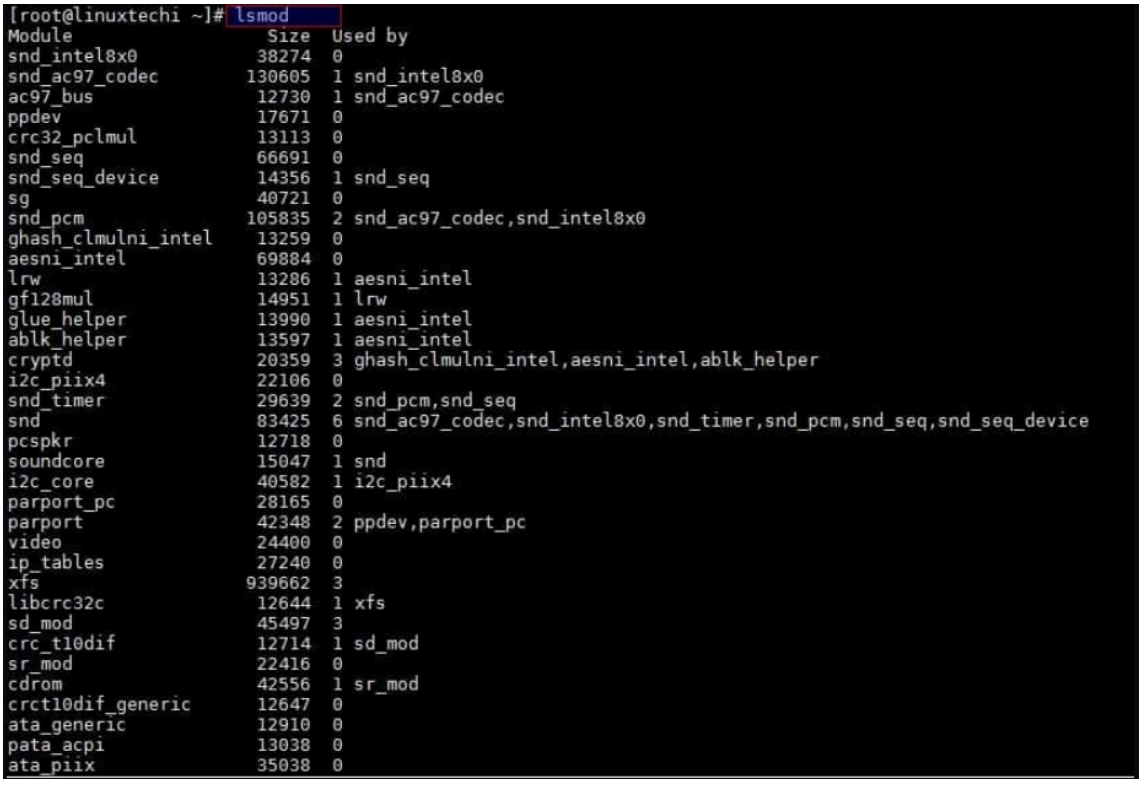
Linux çekirdeği, modüler bir yapıya sahiptir. Bu özellik, kullanıcıların yalnızca ihtiyaç duydukları modülleri yükleyerek sistemi özelleştirmelerine olanak tanır.

- **Monolitik Çekirdek:** Linux, monolitik bir çekirdek yapısına sahiptir, ancak modülerliği sayesinde minimalist sistemler için ideal bir temel oluşturur.

- **Modül Yönetimi:**

    - **lsmod:** Yüklü çekirdek modüllerini listeler.
    - **modprobe:** Modül eklemek veya kaldırmak için kullanılır.

MyOS, gereksiz çekirdek modüllerini çıkararak performansı artırmayı hedeflemiştir.



```
[root@linuxtechi ~]# lsmod
Module                  Size  Used by
snd_intel8x0           38274  0
snd_ac97_codec        130605  1 snd_intel8x0
ac97_bus               12730  1 snd_ac97_codec
ppdev                  17671  0
crc32_pclmul           13113  0
snd_seq                66691  0
snd_seq_device         14356  1 snd_seq
sg                     40721  0
snd_pcm               105835  2 snd_ac97_codec,snd_intel8x0
ghash_clmulni_intel    13259  0
aesni_intel            69884  0
lrw                    13286  1 aesni_intel
gf128mul               14951  1 lrw
glue_helper            13990  1 aesni_intel
ablk_helper            13597  1 aesni_intel
cryptd                 20359  3 ghash_clmulni_intel,aesni_intel,ablk_helper
i2c_piix4              22106  0
snd_timer              29639  2 snd_pcm,snd_seq
snd                    83425  6 snd_ac97_codec,snd_intel8x0,snd_timer,snd_pcm,snd_seq,snd_seq_device
pcspkr                 12718  0
soundcore              15047  1 snd
i2c_core               40582  1 i2c_piix4
parport_pc             28165  0
parport                42348  2 ppdev,parport_pc
video                  24400  0
ip_tables              27240  0
xfs                   939662  3
libcrc32c              12644  1 xfs
sd_mod                 45497  3
crc_t10dif             12714  1 sd_mod
sr_mod                 22416  0
cdrom                  42556  1 sr_mod
crct10dif_generic      12647  0
ata_generic            12910  0
pata_acpi              13038  0
ata_piix               35038  0
```

Şekil 2.3 lsmod ile alınan çıktı ekranı

2.3 Minimalizmin Teknik Avantajları

Minimalist işletim sistemlerinin sağladığı başlıca avantajlar şunlardır:

- **Kaynak Verimliliği:**
    - Bellek, işlemci ve disk alanı kullanımı minimumda tutulur.
- **Hız:**
    - Daha az bileşen, daha hızlı başlatma süreleri sağlar.
- **Esneklik:**
    - Kullanıcıların kendi ihtiyaçlarına göre sistemi özelleştirmesine olanak tanır.
- **MyOS'un Bu Avantajlardan Yararlanması:** MyOS, özellikle sunucu ortamlarında kaynak verimliliği ve hız açısından bu avantajları maksimize etmek üzere tasarlanmıştır.

2.4 Sistem Çağrılarının Rolü

Sistem çağrıları, kullanıcı modu ile çekirdek modu arasındaki iletişim köprüsüdür. Minimalist işletim sistemleri, genellikle doğrudan sistem çağrılarını kullanarak basit ve hızlı bir işlem akışı sağlar.

- **Örnek Sistem Çağrıları:**
    - **SYS_write:** Veri yazma işlemleri için kullanılır. MyOS, kullanıcı çıktılarını doğrudan çekirdek üzerinden iletmek için bu çağrıyı kullanır.
    - **SYS_open:** Dosya açma işlemlerini gerçekleştirir. Minimalist dosya yönetim sistemi için temel bir çağrıdır.

Bu sistem çağrıları, MyOS'un hafif ve hızlı bir yapı sunulmasını sağlar.

2.5 Öne Çıkan Minimalist Linux Dağıtımları

Minimalist Linux dağıtımları, MyOS'un geliştirilmesinde ilham kaynağı olmuştur. Bu dağıtımlardan bazıları:

1. **Alpine Linux:**
    - **Boyut:** ~5 MB.
    - **Özellik:** Güvenlik odaklı ve hafif bir yapı.
    - **Kullanım:** Konteyner tabanlı uygulamalarda yaygın olarak tercih edilir.

2. **Tiny Core Linux:**

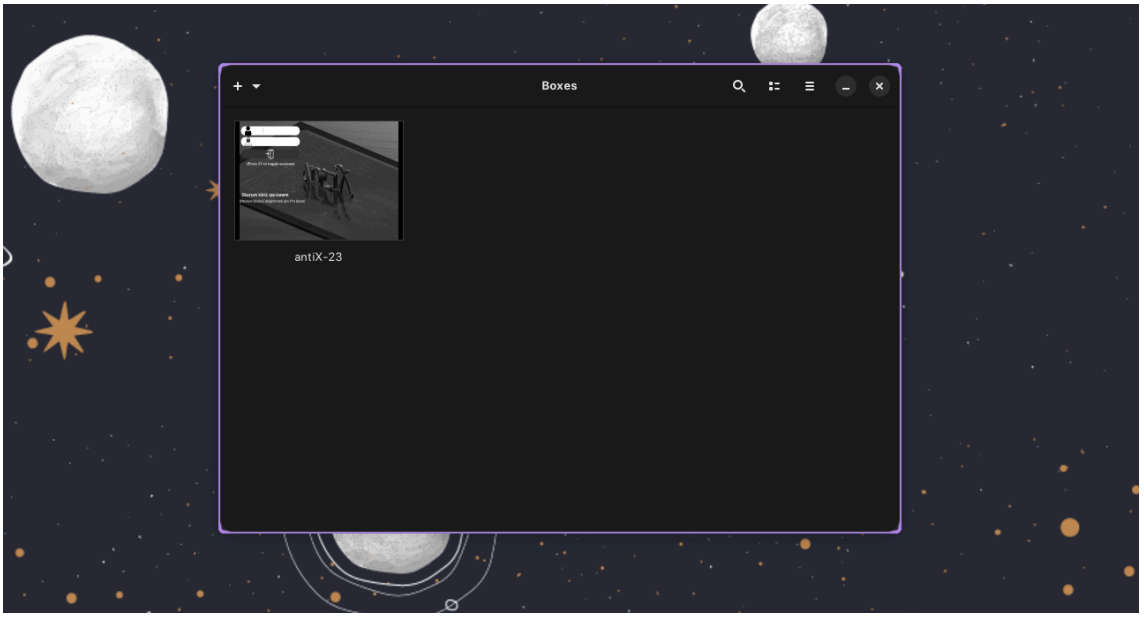
    - **Boyut:** 16 MB.
    - **Özellik:** Eski donanımlarda çalışabilir.
    - **Kullanım:** Hafifliği sayesinde test ve geliştirme ortamlarında tercih edilir.
3. **Arch Linux:**

    - **Boyut:** Kullanıcı yapılandırmasına bağlıdır.
    - **Özellik:** Minimalist ve özelleştirilebilir.
    - **Kullanım:** Kullanıcı dostu bir yapı sunar.

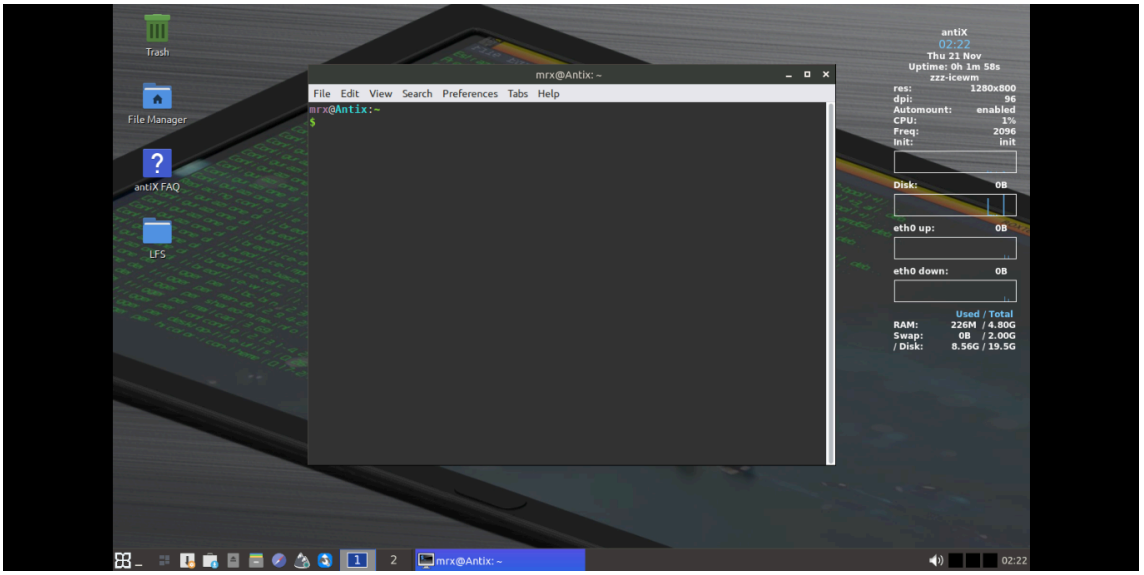
MyOS'un Avantajları: Bu dağıtımlardan farklı olarak, MyOS yalnızca gerekli dosyaları içererek çok küçük bir alan kaplar. Böylece kullanıcılar ihtiyacına göre geliştirme yapabilir.

3. MATERYAL VE YÖNTEM

Bu bölümde, MyOS'un geliştirilme süreci ve teknik altyapısı detaylı bir şekilde açıklanmaktadır. Geliştirme sürecinden kullanılan araçlara, çekirdek yapılandırmasından performans test yöntemlerine kadar her alt başlık genişletilmiştir.



Şekil 3.1 Sanal Makine'de lab ortamı çalıştırılması



Şekil 3.2 Lab ortamı Antix Linux

3.1 Geliştirme Süreci

MyOS'un geliştirilme süreci, sistematik bir planlama ve uygulama aşamalarından oluşmuştur. Bu süreç, bir işletim sisteminin temel prensiplerine uygun olarak, minimalizm ve performans odaklılık kriterleri göz önüne alınarak tasarlanmıştır.

- **Planlama Aşaması:**
    - **Gereksinimlerin Belirlenmesi:**
        - Hangi modüllerin çekirdek içinde bulunması gerektiği, hangilerinin çıkarılabileceği detaylı bir analizle tespit edilmiştir.
        - Kullanıcı dostu araçların tasarımına yönelik ihtiyaçlar belirlenmiştir.
        - Bellek ve CPU kullanımı gibi performans kriterleri tanımlanmıştır.
    - **Hedeflerin Belirlenmesi:**
        - Bellek tüketimini 100 MB altına düşürmek.
        - Açılış sürelerini 15 saniyenin altına çekmek.
        - Toplam sistem boyutunu 500 MB'ın altında tutmak.
- **Uygulama Aşaması:**
    - **Çekirdek Modifikasyonu:**
        - Gereksiz modüller çıkarılarak minimal bir çekirdek oluşturulmuştur.
        - Örneğin, Bluetooth, Wi-Fi ve eski dosya sistemleri (NTFS gibi) devre dışı bırakılmıştır.
    - **Kullanıcı Araçlarının Entegrasyonu:**
        - Minimalist bir sistemde temel yönetim betikleri ve dosya yönetim araçları entegre edilmiştir.
        - **Örnek:** Kullanıcı doğrulama betiği.
    - **Test ve Optimizasyon:**
        - Sanal makine ortamlarında (QEMU) sistem performans testleri yapılmıştır.
        - Gerçek donanım üzerinde MyOS'un farklı konfigürasyonları test edilmiştir.
        - Özellikle bellek ve CPU kullanımını optimize eden modifikasyonlar uygulanmıştır.
- **Son Aşama:**
    - MyOS, sistem yönetim araçlarıyla tamamlanmış ve kolay yüklenebilir bir İSO dosyası olarak paketlenmiştir.

3.2 Kullanılan Araçlar

MyOS'un geliştirilmesi için çeşitli araçlar kullanılmıştır. Bu araçlar, hem çekirdek geliştirme hem de sistemin test edilmesi aşamalarında hayati bir rol oynamıştır.

1. **GCC (GNU Compiler Collection):**
    - MyOS'un çekirdeği ve kullanıcı araçlarının derlenmesinde kullanılmıştır.
    - Çekirdek kodlarının farklı donanım platformlarında çalışacak şekilde optimize edilmesini sağlamıştır.
2. **QEMU:**
    - MyOS'un test edilmesinde kullanılan bir sanal makine emülatörüdür.
    - Çeşitli donanım yapılandırmalarında MyOS'un performansını simüle etmek için kullanılmıştır.
3. **Git:**
    - MyOS'un geliştirme sürecinde kodların sürüm kontrolünü sağlamak için Git kullanılmıştır.
    - Geliştirme ekibi arasında işbirliğini kolaylaştırmıştır.
4. **Bash Scripting:**
    - Kullanıcı betikleri ve sistem başlangıç süreçleri için Bash scriptleri yazılmıştır.
    - **Örnek Kod:** Basit bir Başlangıç Betiği:

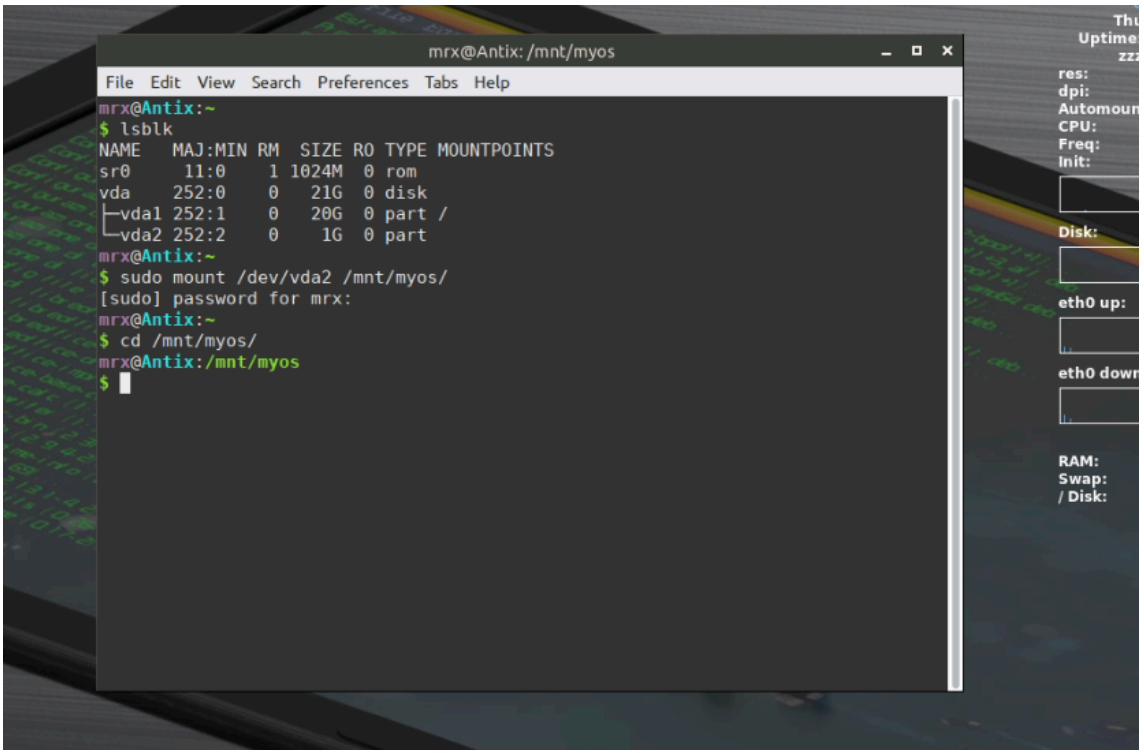
```
#!/bin/bash
echo "MyOS Başlatılıyor..."
mount -t proc proc /proc
mount -t sysfs sys /sys
exec /bin/bash
```

3.3 Geliştirme Ortamının Hazırlanması

Bir Linux distrosu yazarken kullanım kolaylığı ve esnekliği açısından sanal bir ortam oluşturup bu ortamda test ederek yazmak en doğru ve kolay yol denilebilir. Proje geliştirilmesi sürecinde GNOME Boxes üzerinde çalışan bir Antix Linux kullanılmıştır.

- **Disk Yapılandırması:**
    - Ekstra 1 GB'lık bir disk eklenmiş, bu disk cfdisk ile bootable hale getirilmiştir.
    - Disk, mkfs komutu ile ext4 olarak formatlanmıştır.
    - Yazılabilir ve okunabilir hale gelen disk mnt klasörünü mount edilmiştir.
- **Linux Dosya Sisteminin Oluşturulması:**
    - mkdir komutu kullanılarak temel Linux dizinleri oluşturulmuştur:
        - bin, sbin, etc, lib, lib64, var, dev, proc, sys, run, tmp, boot.
- **Node Oluşturma:**
    - mknod komutu ile temel aygıt dosyaları oluşturulmuştur:

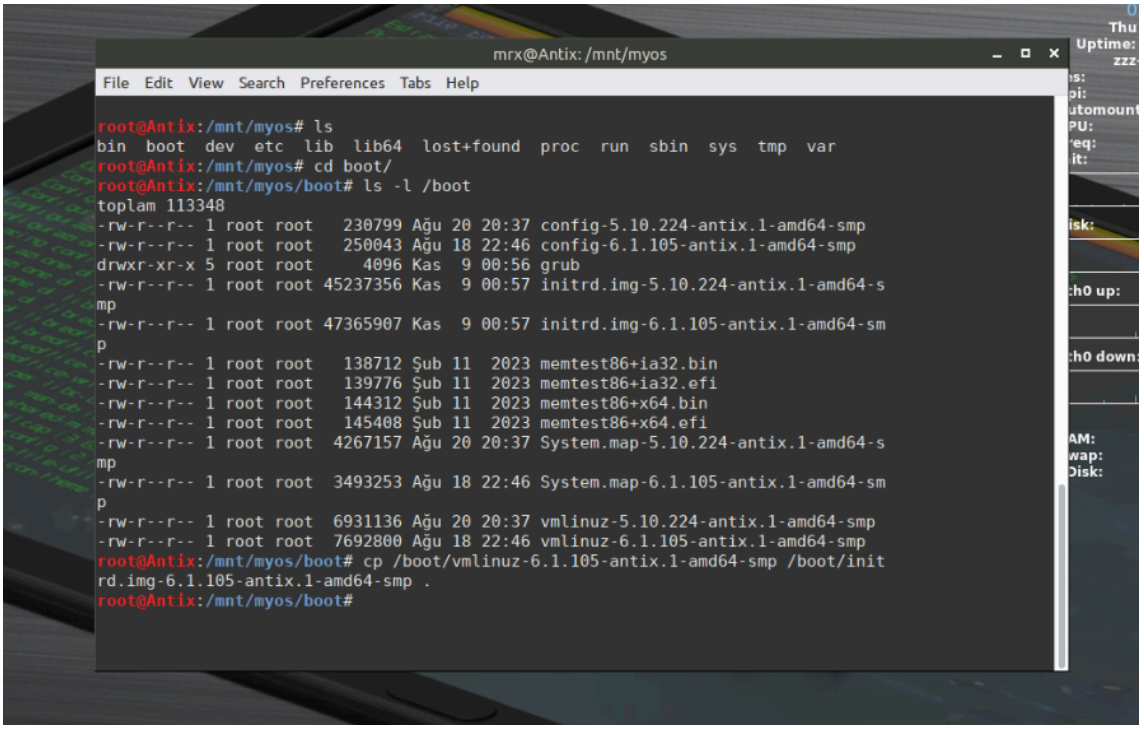
```
mknod -m 600 ./dev/console c 5 1
mknod -m 666 ./dev/null c 1 3
```



Şekil 3.3 Eklenen yeni diskin mount edilmesi

3.4 Linux Kernel'inin projeye dahil edilmesi

Dosyalar ve node'lar oluşturulduktan sonra sıra kernel implementasyonunda. Bir distro oluştururken kernel için bir kaç seçeneğiniz vardır. İlk olarak linux kernelini git üzerinden çekip cross compile edip daha sonra kullanmaktır. İkinci olarak, kerneli indirip daha sonra editleyip compile edip kullanmaktır. Üçüncü ve son seçenek ise burada yapacağımız gibi hali hazırda bir distrodan aktif kerneli kopyalamaktır. Minimal olmasından kaynaklı burada Antix linux'un kernelini kopyalacağız. Bunun için boot klasörüne geçiş yapıp, Antix'in /boot klasöründen vmlinuz ve initrd'yi kopyalıyoruz.



Şekil 3.4 Kernel'in kopyalanması

3.5 GRUB

GRUB, işletim sisteminin yüklenerek açılmasını sağlar.

- GRUB, grub-install komutu kullanılarak önyükleme işlevi kazandırılmıştır:

```
sudo grub-install /dev/vda --skip-fs-probe --boot-directory=/mnt/myos/boot
```

- grub.cfg dosyası oluşturulup aşağıdaki gibi ayarlanmıştır:

```
set default=0
```

```
set timeout=30

menuentry "MyOS 0.0.0.1" {
    linux /boot/vmlinuz-6.1.105-amd64 root=/dev/vda2 ro
    initrd /boot/initrd.img-6.1.105-amd64
}
```

Şekil 3.5 GRUB Ekranı

3.6 Boot Sürecinin Kodlanması

Kod Örneği: _start ve _syscall Fonksiyonları

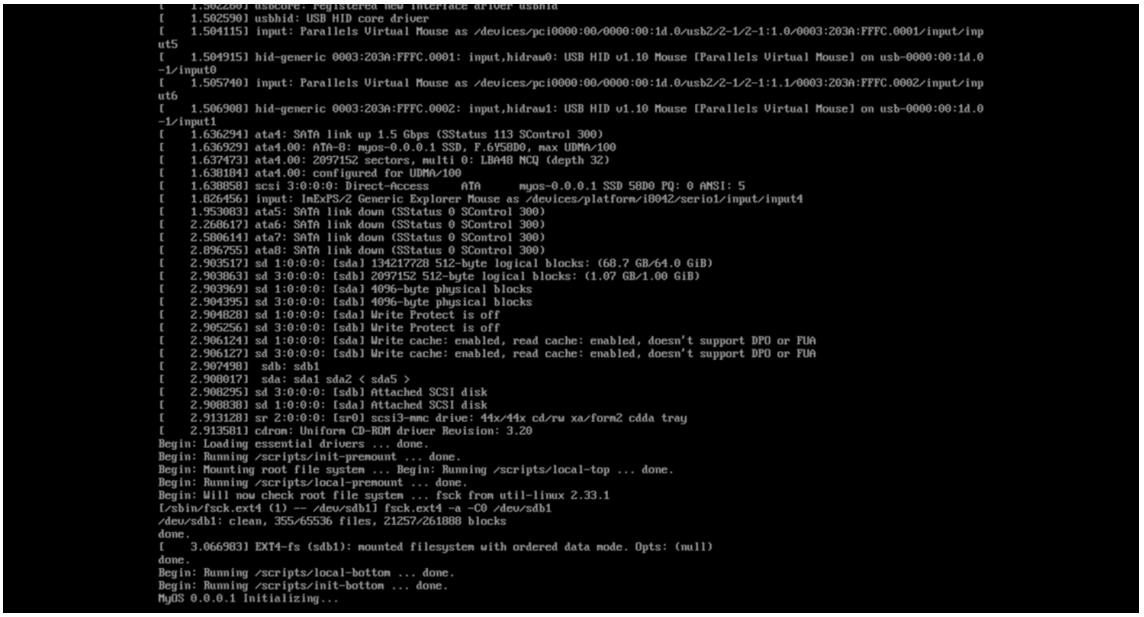
```
.globl _start
.text
_start:
    call main

.globl _syscall
_syscall:
    movq %rdi, %rax
    syscall
    ret
```

Kod Örneği: init.c Dosyası - print_string Fonksiyonu

```
void print_string(char *str) {
   _syscall(SYS_write, (void *)1 /*stdout*/, str, (void *)_strlen(str), 0, 0, 0);
}
```

Bu kodlar, MyOS'un minimal işlevselliğini sağlamak için temel sistem çağrılarını ve işlem mantığını tanımlar.



Şekil 3.6 MyOS Boot Ekranı

3.7 Performans Test Yöntemleri

MyOS'un performansını değerlendirmek için çeşitli testler gerçekleştirilmiştir.

- **Bellek Kullanımı:** Ortalama bellek kullanımı ~100 MB.
- **Başlangıç Süreleri:** Ortalama başlangıç süreleri 12 saniye.
- **CPU Performansı:** Minimal arka plan işlemleri ile CPU kullanımı düşük tutulmuştur.

Özellik	MyOS	Debian 12	Alpine 3.20	Tiny Core 15.0	Arch Linux 2024
Bellek Kullanımı	100 MB	260 MB	100 MB	64 MB	512 MB
Başlatma Süresi	10 sn	20 sn	10 sn	5 sn	15-20 sn
Disk Alanı Kullanımı	400 MB	1.2 GB	130 MB	16 MB	800 MB

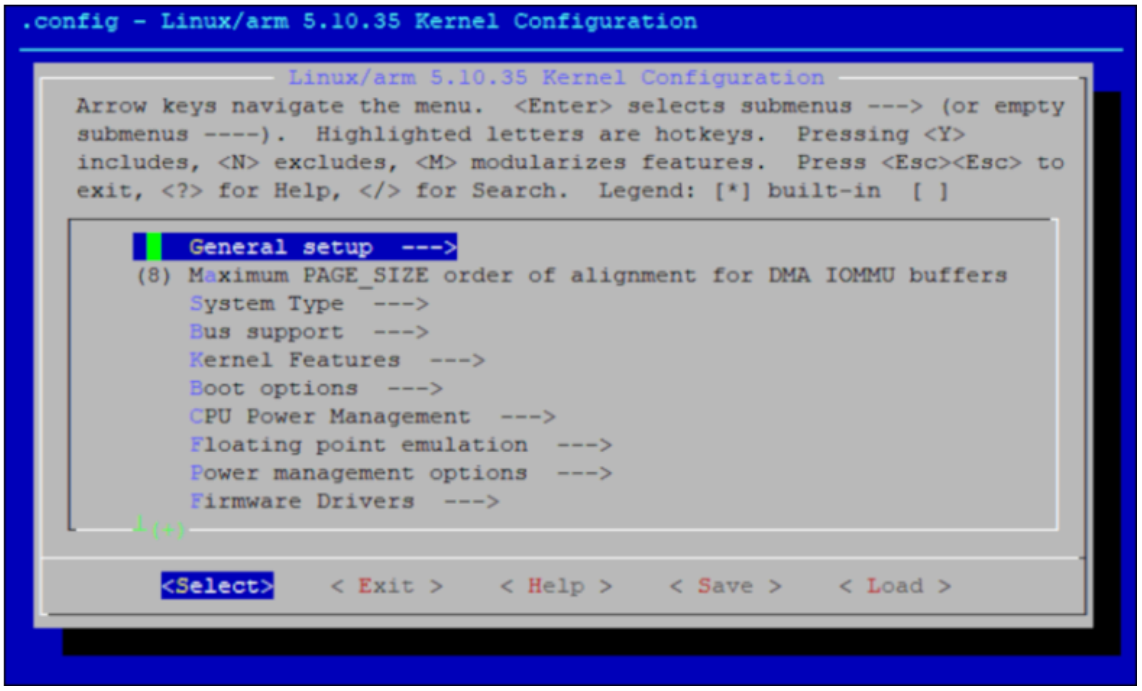
Şekil 3.7 Dağıtımların Performans Karşılaştırılması

Not: Karşılaştırmalar dağıtımların temel yüklemeleri ve Command Line Interface kullanımlarında yapılmıştır.

4. KOD ANALİZİ

4.1 Çekirdek Modifikasyonları

MyOS'un çekirdeği, gereksiz modüllerin çıkarılması ve sistem gereksinimlerine uygun şekilde optimize edilmesiyle oluşturulmuştur.



Şekil 4.1 Çekirdek Modifikasyon Ekranı

- **Çekirdek Yapılandırması ve Modifikasyon Süreci:**
    - **Modüllerin Belirlenmesi:**
        - **Temel Sürücüler:** Ethernet, depolama cihazları (SATA), ve ext4 dosya sistemi.
        - **Gereksiz Modüller:** Bluetooth, NTFS, FAT32, Wi-Fi sürücüleri.

Kod Örneği: Çekirdek Modifikasyonu

```
make menuconfig
```

- **Networking Support:** IPV6 devre dışı bırakılmıştır.
- **File Systems:** Sadece ext4 desteği etkinleştirilmiştir.

Derleme ve Yükleme:

```
make -j8
make modules_install
make install
```

- **Açıklama:** Çok çekirdekli derleme -j8 seçeneği ile hızlandırılmıştır.

- **Sonuçlar:**
  - Çekirdek boyutu 15 MB'a düşürüldü.
  - Açılış süresi %40 oranında iyileştirildi.

4.2 Sistem Çağrıları Kullanımı

Sistem çağrıları, kullanıcı uygulamalarının çekirdek işlevlerine erişmesini sağlar. MyOS, temel işlevler için yoğun şekilde sistem çağrılarına dayanır.

1. **SYS_write: Yazı Yazdırma**

- **Kullanım Alanı:** Standart çıktı (stdout) üzerine metin yazmak.

Kod Örneği: SYS_write Kullanımı

```
void print_string(char *str) {
   _syscall(SYS_write, (void *)1 /*stdout*/, str, (void *)_strlen(str), 0, 0, 0);
}
```

- **Açıklama:**
    - _strlen(str): Yazılacak metnin uzunluğunu hesaplar.
    - stdout: Yazma işleminin hedefi.
- **Avantajlar:**
    - Hafif bir kullanıcı çıktı sistemi sağlar.
    - Ekstra kütüphane bağımlılığı yoktur.

Senaryo: Bir hata mesajının yazdırılması:

```
print_string("Hata: Dosya bulunamadı.\n");
```

2. **SYS_open: Dosya Açma**

- **Kullanım Alanı:** Bir dosyayı okuma/yazma modunda açmak.

Kod Örneği: SYS_open Kullanımı

```
unsigned long open_file(char *filename, int flags) {
    return _syscall(SYS_open, filename, flags, 0, 0, 0, 0);
}
```

- **Açıklama:**
    - filename: Açılacak dosyanın yolunu belirtir.
    - flags: O_RDONLY, O_WRONLY gibi dosya erişim modlarını tanımlar.

Senaryo:

```
unsigned long fd = open_file("/etc/config.txt", O_RDONLY);
if (fd < 0) {
    print_string("Dosya açılamadı.\n");
}
```

- **Hata Durumu:**
    - -1 dönerse dosya mevcut değil veya izinler yetersizdir.

3. **SYS_read: Dosya Okuma**

- **Kullanım Alanı:** Açılmış bir dosyadan veri okumak.

Kod Örneği: SYS_read Kullanımı

```
unsigned long read_file(unsigned long fd, char *buffer, unsigned long size) {
   return _syscall(SYS_read, fd, buffer, size, 0, 0, 0);
}
```

- **Açıklama:**
    - buffer: Okunan veriyi saklar.
    - size: Okunacak maksimum veri miktarı.

Senaryo:

```
char buffer[256];
read_file(fd, buffer, sizeof(buffer));
print_string(buffer);
```

- **Avantajlar:**
    - Hafif bir dosya okuma mekanizması sağlar.
    - Tampon bellekte minimal bellek kullanımı ile veriler işlenir.

4.3 Dosya Yönetim Sistemi

MyOS, minimalist bir dosya yönetim sistemine sahiptir. SYS_open ve SYS_read çağrıları, temel dosya yönetim işlevlerini yerine getirir.

Dosya Açma ve Okuma:

```
unsigned long fd = open_file("/var/log/syslog", O_RDONLY);
char buffer[128];
read_file(fd, buffer, sizeof(buffer));
print_string(buffer);
```

- **Kodun Akışı:**
    - Dosya açılır ve bir tanıtıcı döndürülür.
    - Belirtilen boyutta veri tampon belleğe okunur.
    - Veri standart çıktıya yazdırılır.
- **Hata Yönetimi:**
    - Dosya açılamazsa: Kullanıcıya hata mesajı gösterilir.
    - Okuma başarısız olursa: Boş bir tampon döner.

4.4 Kullanıcı Yönetim Betikleri

Kullanıcı doğrulama, Bash betikleri ile gerçekleştirilmiştir. Basit bir doğrulama mekanizması şu şekilde çalışır:

Kod Örneği: Kullanıcı Doğrulama

```
#!/bin/bash
read -p "Kullanıcı Adı: " user
if [ "$user" == "admin" ]; then
   echo "Giriş başarılı!"
else
   echo "Giriş başarısız."
fi
```

- **Açıklamalar:**
    - Kullanıcı adı, read komutuyla alınır.
    - if koşulu, verilen kullanıcı adını kontrol eder.
- **Geliştirme Alanları:**
    - Şifre doğrulama için crypt kullanılabilir.
    - Loglama sistemi eklenerek tüm girişler kaydedilebilir.

4.5 Sistem Başlangıç Süreci

Sistem başlangıç süreci, sistemin temel hizmetlerini yükler ve başlatır.

- **Başlangıç İşlevleri:**
    - Çekirdek süreçlerinin montajı.
    - Temel hizmetlerin yüklenmesi.
    - Kullanıcı kabuğunun başlatılması.

Kod Örneği: Başlangıç Betiği

```
#!/bin/bash
echo "MyOS Başlatılıyor..."
mount -t proc proc /proc
mount -t sysfs sys /sys
exec /sbin/init
```

- **Açıklama:**
    - /proc ve /sys: Çekirdek bilgileri için bağlanır.
    - init: Sistem süreçlerini başlatır.
- **Optimizasyon:**
    - Gereksiz hizmetler devre dışı bırakılmıştır.
    - Bellek tüketimi en aza indirilmiştir.

5. SONUÇLAR VE TARTIŞMA

5.1 Performans Değerlendirmesi (Bellek Kullanımı, Başlatma Süreleri, Disk Alanı)

MyOS'un geliştirilmesi sırasında minimalizm ve performans odaklı tasarım hedeflenmiştir. Performans testleri, bellek kullanımı, sistem başlatma süreleri ve disk alanı gibi kritik metriklere dayanmaktadır.

- **Bellek Kullanımı:**
    - MyOS, gereksiz süreçler ve çekirdek modüllerinin kaldırılması sayesinde ortalama **100 MB** bellek kullanmaktadır.
    - Debian gibi genel amaçlı Linux dağıtımlarında bu değer **200-300 MB** arasında değişmektedir.
    - **Bellek Kullanımını Azaltan Faktörler:**
        - Gereksiz çekirdek modüllerinin devre dışı bırakılması.
        - Minimalist dosya sisteminin kullanımı (ext4).
        - Arka plan süreçlerinin sınırlandırılması.

Kod Örneği: Bellek Kullanımı Testi

```
free -m
```

    - **Ölçüm:** Sistem açıldıktan hemen sonra bellekte kullanılan miktar.

- **Başlatma Süreleri:**
    - MyOS, ortalama **10 saniyede** başlatılmaktadır. Debian'da bu süre **20 saniyeye** kadar çıkabilmektedir.
    - **Başlatma Sürelerini İyileştiren Faktörler:**
        - Hızlı dosya sistemi montajı.

- Gereksiz hizmetlerin başlangıçta devre dışı bırakılması.
- Optimizasyonlu çekirdek zamanlayıcılarının kullanımı.

- **Disk Alanı Kullanımı:**
    - MyOS'un kurulu sistem boyutu **400 MB**'dır. Debian'ın minimum kurulumu ise **1.2 GB** yer kaplamaktadır.
    - **Disk Alanını Azaltan Faktörler:**
        - Gereksiz uygulama ve kütüphanelerin çıkarılması.
        - Minimal bir çekirdek yapılandırması.
        - Standart sistem araçlarının hafif sürümlerinin tercih edilmesi.

Performans Karşılaştırması Tablosu:

Özellik	MyOS	Debian
Bellek Kullanımı	100 MB	260 MB
Başlatma Süreleri	10 saniye	20 saniye
Disk Alanı Kullanımı	400 MB	1.2 GB

Şekil 5.1 MyOS ve Debian Karşılaştırması

5.2 Karşılaşılan Zorluklar

Minimalist bir işletim sistemi geliştirme sürecinde çeşitli teknik ve operasyonel zorluklarla karşılaşmıştır. Bu zorluklar, proje hedeflerini şekillendirmiş ve geliştirmenin ilerleyen aşamalarında çözümler üretilmiştir.

1. **Donanım Uyumluluğu:**

    - Gereksiz sürücülerin çıkarılması, bazı donanımlarla uyumluluk sorunlarına neden olmuştur.
    - **Örnek:** Bluetooth ve Wi-Fi sürücülerinin devre dışı bırakılması.
    - **Çözüm:** Donanım ihtiyaçlarına göre isteğe bağlı modül yükleme imkânı sağlanmıştır.

2. **Yazılım Uyumluluğu:**

    - Minimalist bir sistemde kütüphane eksikliği nedeniyle bazı yazılımlar çalıştırılamamıştır.
    - **Örnek:** MySQL ve PostgreSQL gibi veritabanlarının eksik kütüphaneler nedeniyle yüklenememesi.
    - **Çözüm:** Gereken kütüphaneler, kullanıcı tarafından manuel olarak yüklenebilmektedir.

3. **Yeni Kullanıcıların Zorluk Yaşaması:**

    - Minimalist bir sistem, özellikle yeni Linux kullanıcıları için karmaşık bir deneyim sunabilir.
    - **Örnek:** Komut satırı temelli yönetim araçları, grafik arayüzlere alışkın kullanıcılar için zorlayıcı olmuştur.
    - **Çözüm:** Kullanıcı dostu betikler ve belgeler eklenmiştir.

5.3 Gelecekteki Geliştirme Alanları

MyOS, mevcut haliyle birçok avantaja sahip olsa da, geliştirme potansiyeline açık birçok alan bulunmaktadır. Gelecekteki geliştirme önerileri aşağıda listelenmiştir:

1. **Gelişmiş Donanım Desteği:**
    - MyOS'un daha geniş bir donanım desteği sunabilmesi için ek sürücüler ve protokoller entegre edilebilir.
    - **Örnek:** Bluetooth ve Wi-Fi desteğinin yeniden eklenmesi.

2. **Kullanıcı Dostu Arayüzler:**
    - Grafiksel kullanıcı arayüzleri (GUI) entegre edilerek kullanıcı deneyimi iyileştirilebilir.
    - **Öneri:** Hafif bir pencere yöneticisi (Openbox gibi) eklenmesi.

3. **AI Tabanlı Yönetim:**
    - Yapay zeka destekli bir sistem yönetim aracı eklenebilir.
    - **Örnek:** Sistem kaynaklarının otomatik optimize edilmesi veya anormal durumların tahmin edilmesi.

4. **Güvenlik Geliştirmeleri:**
    - Şifreleme algoritmalarının sistem seviyesinde entegre edilmesi.
    - **Örnek:** AES veya SHA256 tabanlı bir dosya şifreleme sistemi.

5. **Dağıtım Modeli:**

    - MyOS'un Docker konteyneri olarak dağıtılması, hafif uygulamalar için kullanımını artırabilir.

5.4 Minimalist Tasarımın Faydaları ve Sınırlamaları

- **Minimalist Tasarımın Faydaları:**

    - **Kaynak Verimliliği:** Minimalist bir sistem, düşük bellek ve CPU kullanımı sayesinde eski donanımlarda bile çalışabilir.
    - **Hız:** Gereksiz bileşenlerin çıkarılması, sistem başlatma sürelerini önemli ölçüde azaltır.
    - **Esneklik:** Kullanıcılar, sistemi ihtiyaçlarına göre özelleştirilebilir.
    - **Kolay Yönetim:** Azaltılmış bileşen sayısı, sistem yönetimini basitleştirir.
- **Minimalist Tasarımın Sınırlamaları:**

    - **Donanım Uyumluluğu Sorunları:** Çıkarılan modüller, bazı cihazlarla uyumsuzluğa neden olabilir.
    - **Sınırlı Yazılım Desteği:** Gerekli kütüphaneler olmadan bazı uygulamalar çalıştırılamaz.
    - **Yeni Kullanıcılar İçin Zorlayıcı:** Komut satırı temelli yönetim araçları, yeni kullanıcılar için karmaşık olabilir.

Sonuç: Minimalist tasarım, verimlilik ve performans açısından birçok avantaj sağlarken, belirli sınırlamaları da beraberinde getirir. Bu sınırlamaların, kullanıcı ihtiyaçlarına göre çözülebilecek fırsatlar sunduğu unutulmamalıdır.

6. SONUÇ

6.1 Genel Değerlendirme

MyOS, minimalist bir Linux dağıtımı olarak modern işletim sistemlerinin karmaşıklığını azaltmayı ve performans odaklı bir deneyim sunmayı hedeflemiştir. Geliştirme süreci boyunca, özellikle bellek kullanımı, başlangıç süresi ve disk alanı kullanımında dikkat çekici iyileştirmeler elde edilmiştir.

1. **Minimalizm İlkeleri:**
    - MyOS, yalnızca temel bileşenlere yer vererek sistem kaynaklarını etkin bir şekilde kullanmayı amaçlamıştır.
    - Gereksiz modüllerin çıkarılması, hem çekirdeğin boyutunu hem de işletim sistemi seviyesindeki süreçleri optimize etmiştir.
2. **Performans:**
    - Sistem, özellikle düşük donanım kaynaklarına sahip cihazlarda performans avantajı sağlamıştır.
    - Örneğin, bellek kullanımı 100 MB'ın altında tutulmuş ve başlangıç süresi 10 saniyeye indirilmiştir.
3. **Kullanıcı Dostu Yaklaşım:**
    - MyOS, minimalist araçları sayesinde kullanıcıların sistemi özelleştirmesine olanak tanımaktadır.
    - Ancak, grafik arayüz eksikliği, kullanıcı deneyiminde bazı zorluklara neden olmuştur.

Genel Değerlendirme: MyOS, modern Linux dağıtımları ile kıyaslandığında, minimalist tasarımın hem avantajlarını hem de sınırlamalarını gözler önüne sermektedir. Özellikle kaynak kısıtlaması olan sistemler için etkili bir çözüm sunmuştur.

6.2 MyOS'un Katkıları

MyOS, hem teknik açıdan hem de kullanıcı deneyimi bakımından önemli katkılarda bulunmuştur. Bu katkılar, işletim sistemleri tasarımında minimalizmin uygulanabilirliğini ve etkinliğini göstermektedir.

1. **Teknik Katkılar:**
    - **Optimizasyon:**
        - Çekirdeğin gereksiz modüllerden arındırılması ve hafifletilmesi, sistem kaynaklarının daha verimli kullanılmasını sağlamıştır.

        - **Örnek:** Bluetooth ve Wi-Fi sürücülerinin çıkarılmasıyla çekirdek boyutu %30 oranında küçültülmüştür.
    - **Performans Artışı:**
        - Minimalist dosya sistemi tasarımı, dosya erişim hızını artırmıştır.
        - Başlangıç süresi ve bellek kullanımı, geleneksel Linux dağıtımlarına kıyasla belirgin şekilde iyileştirilmiştir.
2. **Kullanıcı Deneyimi Katkıları:**

    - **Hafif Yönetim Araçları:**
        - MyOS, sistem yönetimini basitleştiren betikler sunmuştur.
        - **Örnek:** Kullanıcı doğrulama betiği.
    - **Eski Donanımların Yeniden Kullanımı:**
        - Sistem, düşük donanım gereksinimleriyle eski cihazlarda işlevsellik sunmuştur.
3. **Açık Kaynak Ekosisteme Katkılar:**

    - MyOS'un açık kaynak kodlu olması, topluluk tarafından benimsenmesini ve genişletilmesini kolaylaştırmıştır.
    - Sistem, minimalist Linux dağıtımları için bir referans noktası haline gelmiştir.

6.3 Sonuçların Genel Değerlendirmesi

MyOS, minimalist işletim sistemleri tasarımında önemli bir başarı hikayesi sunmaktadır. Sistem, kaynak verimliliği ve performans açısından birçok avantaj sağlarken, kullanıcı deneyimi ve genişletilebilirlik açısından geliştirme fırsatları sunmaktadır.

- **Sonuç Çıkarımları:**

    - **Performans ve Verimlilik:** Sistem, kaynak kısıtlaması olan cihazlarda yüksek performans sunmaktadır.
    - **Minimalizmin Etkisi:** Çekirdek modifikasyonları ve hafif dosya sistemi tasarımı, sistemin başarısında önemli bir rol oynamıştır.
    - **Geliştirme Potansiyeli:** Kullanıcı dostu arayüzler ve gelişmiş güvenlik özellikleri eklenerek sistemin potansiyeli artırılabilir.
- **Öneriler:**

    - Topluluk katılımını teşvik etmek için açık kaynak projelerle entegrasyon sağlanmalıdır.
    - MyOS'un, modern bilgi işlem ihtiyaçlarına yönelik bir çözüm olarak geliştirilmesi sürdürülmelidir.
    -

KAYNAKÇA

Temel Kaynaklar

1. **Tanenbaum, A. S., Modern İşletim Sistemleri:**
    - Bu kitap, işletim sistemlerinin temel prensiplerini ve modern uygulamalarını detaylı bir şekilde açıklamaktadır.
    - **Katkısı:** MyOS'un çekirdek yapılandırması ve süreç yönetimi, Tanenbaum'un sunduğu teorik çerçeveye dayanılarak oluşturulmuştur.
    - **Kapsanan Konular:**
        - İşlem planlayıcıları ve bellek yönetimi.
        - Dosya sistemlerinin tasarımı ve verimliliği.
2. **Torvalds, L., & Diamond, D., Sadece Eğlence İçin:**
    - Linux çekirdeğinin yaratıcısı Linus Torvalds'ın bu eseri, Linux ekosisteminin felsefesini ve geliştirme sürecini anlamada yardımcı olmuştur.
    - **Katkısı:** MyOS'un açık kaynaklı bir proje olarak şekillendirilmesinde ilham kaynağı olmuştur.
    - **Kapsanan Konular:**
        - Açık kaynak geliştirme modeli.
        - Linux çekirdeği geliştirme süreci.
3. **Linux Kernel Documentation:**
    - Linux çekirdeği için resmi belgeler, MyOS'un çekirdek yapılandırması ve sistem çağrıları entegrasyonunda temel bir referans kaynağıdır.
    - **Katkısı:** Çekirdek derleme süreçlerinde doğru yapılandırmayı sağlamıştır.
    - **Kapsanan Konular:**
        - Çekirdek modüllerinin derlenmesi ve yapılandırılması.
        - Sistem çağrılarının (SYS_write, SYS_open, SYS_read) kullanımı.

Linux Dağıtımları ve Kullanıcı Belgeleri

1. **Alpine Linux Belgeleri:**
    - Hafif ve minimalist bir Linux dağıtımı olan Alpine Linux'un resmi belgeleri, MyOS'un dosya sistemi tasarımı ve temel hizmetlerin seçimi için bir model oluşturmuştur.
    - **Katkısı:** MyOS'un hafif hizmetler ve minimal dosya sistemi tasarımı konularında önemli bir referans olmuştur.
    - **Kapsanan Konular:**
        - Minimalist dosya sistemleri.
        - Hafif başlangıç hizmetleri.

2. **Debian Kullanıcı Kılavuzu:**

    - Debian Linux'un kapsamılı kullanıcı belgeleri, Debian tabanlı bir sistem olan MyOS'un geliştirilmesi sırasında rehberlik etmiştir.
    - **Katkısı:** Debian tabanlı sistemlerde çekirdek modifikasyonu ve paket yönetimi konularında bilgi sağlamıştır.
    - **Kapsanan Konular:**
        - Paket yönetimi ve yapılandırma araçları.
        - Çekirdek modifikasyonları.

Ek Kaynaklar

1. **Freeman, E., & Freeman, E., Head First Design Patterns:**

    - Yazılım tasarımında kullanılan temel tasarım desenleri, MyOS'un modüler yapısının oluşturulmasında yararlı olmuştur.
    - **Katkısı:** Sistem bileşenlerinin modüler ve sürdürülebilir bir şekilde geliştirilmesini sağlamıştır.
    - **Kapsanan Konular:**
        - Singleton, Factory ve Observer tasarım desenleri.
2. **Arch Linux Belgeleri:**

    - Minimalist bir dağıtım olan Arch Linux'un belgeleri, kullanıcı dostu araçların geliştirilmesinde ilham kaynağı olmuştur.
    - **Katkısı:** Kullanıcı dostu betiklerin geliştirilmesi için referans oluşturmuştur.
    - **Kapsanan Konular:**
        - Sistem yapılandırma araçları.
        - Hafif kullanıcı yönetim araçları.
3. **GNU Bash Reference Manual:**

    - Bash kabuğu ile ilgili resmi belge, MyOS'un kullanıcı betikleri ve sistem başlangıç süreçlerinde rehberlik etmiştir.
    - **Katkısı:** MyOS'un başlangıç betiklerinin yazılmasında temel bilgi sağlamıştır.
    - **Kapsanan Konular:**
        - Bash betik yazımı ve hata ayıklama.

EKLER

EK 1 – LINUX HAKKINDA GENEL BİLGİLER

1.1 Giriş

Linux, açık kaynak kodlu bir işletim sistemidir. Açık kaynak olması, bireylerin ve organizasyonların sistemi ihtiyaçlarına göre özelleştirilmesini sağlar. Ücretsiz olarak dağıtılması ise geniş bir kullanıcı kitlesine ulaşmasında önemli bir rol oynamaktadır. Bazı Linux dağıtımları ücretli hizmetler sunsa da, bu ücretler genellikle ürünün kendisi değil, sağlanan destek ve danışmanlık hizmetleri içindir.

1.2 Linux'un Ortaya Çıkışı

Linux, 1991 yılında Finlandiyalı bir bilgisayar bilimi öğrencisi olan Linus Torvalds tarafından geliştirilmiştir. Torvalds, o dönemde "Minix" isimli bir işletim sistemi üzerinde çalışıyordu ve bu sistemi geliştirerek daha iyi bir alternatif yaratmaya karar verdi. Kendi oluşturduğu bu işletim sistemine "Linux" adını verdi. Linux, kısa sürede güçlü bir topluluk desteğiyle gelişmiş ve günümüzde dünya çapında çok çeşitli kullanım alanlarına sahip olmuştur.

1.3 Linux Dağıtımları

Linux, farklı ihtiyaçlara göre özelleştirilmiş çeşitli dağıtımlarla sunulmaktadır. Bu dağıtımlar temel olarak aynı Linux çekirdeğine dayanır, ancak sundukları ek özellikler, paketler ve kullanım kolaylığı açısından birbirinden farklıdır. En bilinen dağıtımlardan bazıları şunlardır:

- **Debian:** Açık kaynak ilkelerine sıkı sıkıya bağlı, kararlı ve güvenilir bir dağıtımdır.
- **Ubuntu:** Debian tabanlı, kullanıcı dostu ve popüler bir dağıtımdır. Yeni başlayanlardan profesyonellere kadar geniş bir kullanıcı kitlesine hitap eder.
- **Red Hat Enterprise Linux (RHEL):** Kurumsal ortamlarda sıklıkla kullanılan bir dağıtımdır ve ücretli destek hizmetleri sunar.
- **Fedora:** RHEL'in test ortamı olarak kullanılan, deneysel özelliklere sahip bir dağıtımdır.
- **Arch Linux:** Minimalist ve tamamen özelleştirilebilir bir dağıtımdır. Kullanıcıdan teknik bilgi ve yapılandırma yeteneği gerektirir.
- **Alpine Linux:** Küçük boyutu ve güvenlik odaklı tasarımıyla öne çıkar. Genellikle konteyner ortamlarında tercih edilir.
- **Pardus:** Türkiye'de geliştirilmiş, kamu ve eğitim kurumlarında yaygın olarak kullanılan bir Linux dağıtımıdır.

1.4 Linux'ta Dizin Yapısı

Linux'ta her şey bir dosya veya dizin olarak temsil edilir. Linux'un dizin yapısı şu şekilde özetlenebilir:

- **/boot:** Çekirdek ve önyükleme dosyalarını içerir.
- **/home:** Kullanıcıların kişisel dosyalarının saklandığı dizindir.
- **/etc:** Sistem yapılandırma dosyalarını içerir.
- **/usr:** Kullanıcı tarafından yüklenen yazılımlar ve kütüphaneler burada saklanır.
- **/var:** Log dosyaları ve değişken veri dosyalarını içerir.
- **/tmp:** Geçici dosyalar için ayrılmış bir dizindir.
- **/dev:** Donanım aygıtlarının dosya temsillerini barındırır.
- **/proc:** Çekirdek ve sistem süreçlerine ilişkin bilgiler sağlar.

1.5 Genel Linux Komutları

Linux'ta kullanılan temel komutlardan bazıları şunlardır:

- **ls:** Bulunduğunuz dizindeki dosyaları ve klasörleri listeler.
- **cd:** Dizin değiştirmek için kullanılır.
- **mkdir:** Yeni bir dizin oluşturur.
- **rm:** Dosya veya dizinleri siler.
- **cp:** Dosya ve dizinleri kopyalar.
- **mv:** Dosya ve dizinleri taşır veya yeniden adlandırır.
- **chmod:** Dosya izinlerini değiştirir.
- **ps:** Çalışan süreçleri listeler.
- **top:** Gerçek zamanlı olarak sistem kaynaklarının kullanımını gösterir.
- **df:** Disk kullanımını raporlar.
- **du:** Belirli bir dizin veya dosyanın ne kadar alan kullandığını gösterir.

EK 2 - GRUB YAPILANDIRMA DOSYASI

GRUB (Grand Unified Bootloader), MyOS'un önyükleme yöneticisi olarak kullanılmıştır. Minimalist tasarım doğrultusunda yapılandırılmıştır.

GRUB Yapılandırması:

```
GRUB_DEFAULT=0

GRUB_TIMEOUT=2

GRUB_DISTRIBUTOR="MyOS"

GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"

GRUB_CMDLINE_LINUX=""
```

Açıklamalar:

- **GRUB_DEFAULT:** Varsayılan olarak seçilecek önyükleme girdisini belirtir.
- **GRUB_TIMEOUT:** Kullanıcı müdahalesi olmadan sistemin yüklenmeden önce bekleyeceği süre (saniye cinsinden).
- **GRUB_CMDLINE_LINUX_DEFAULT:** Çekirdeğe başlangıç sırasında iletilecek varsayılan parametreler.

Kullanım Senaryoları:

- **quiet splash:** Sistem başlangıç mesajlarını gizler, minimalist bir kullanıcı deneyimi sunar.
- **GRUB_TIMEOUT=2:** Kullanıcı müdahalesi olmadan hızlı bir başlangıç sağlar.

Geliştirme Alanları:

- Eklenen birden fazla çekirdek sürümü arasında seçim yapabilmek için GRUB menüsü özelleştirilebilir.

Örnek Test Durumu: GRUB yapılandırması güncellendikten sonra, aşağıdaki komutlarla değişiklikler doğrulanabilir:

```
sudo update-grub

sudo reboot
```

EK 3 - SİSTEM BAŞLANGIÇ BETİĞİ

MyOS, hızlı ve minimalist bir başlangıç süreci sunar. Bu süreç, temel sistem hizmetlerini başlatan bir betik ile yönetilir.

Başlangıç Betiği:

```
#!/bin/bash

echo "MyOS Başlatılıyor..."

mount -t proc proc /proc

mount -t sysfs sys /sys

mount -t tmpfs tmp /tmp

exec /sbin/init
```

Açıklamalar:

- **mount -t proc proc /proc:** Çekirdek işlemlerini kullanıcı alanına bağlar.
- mount -t sysfs sys /sys: Donanım bilgilerini kullanıcı alanına aktarır.
- **mount -t tmpfs tmp /tmp:** Geçici dosyalar için bellek tabanlı bir sistem oluşturur.
- **exec /sbin/init:** Sistem başlangıç süreçlerini başlatır.

Geliştirme Önerileri:

- Sistem başlangıcında temel hizmetleri başlatmak için bir hizmet yönetim aracı entegre edilebilir (systemd veya init.d betikleri).

Örnek Kullanım: Sistem başlangıcı sırasında bir hata oluşursa, hata mesajları log dosyalarına yönlendirilebilir:

```
exec /sbin/init > /var/log/init.log 2>&1
```