

# STAJ BAVURUSU

Mehmet Ali Akmanalp  
(makmanalp@wpi.edu)

6 Nisan 2008

../Downloads/C:/Users/infralite/Pictures/adasd.jpg

## 1 Önsöz

Bu yaz size biraz fazla uzunmu gibi gelebilir, fakat sizi temin ederim ki deil! Ben-den kendim hakkında yeterli bilgi salamam istendi, ve ben de (çok skc olmamasna dikkat ederek, fakat profesyonellii de hepten bovermeden) neyin cidden gerekli olduuna karar verdiğem ekledim. Üstelik burada aratrarak bulabileceinizden çok daha fazlas var, kendim merak edip denedim de. e almay düündüğünüz birini tanyp deerlendirirken harcayacanz fazladan birkaç dakikamn, kararnz ne olursa olsun, sonradan size kat kat geri döneceine inanyorum. Umarm bu kadar kendimi size en iyi ekilde tantmama yeter.

## 2 Ben Kimim?

1989 yılında stanbul’da dodum. Her eyin içini açan meraklı bir çocuk olarak büyüdüm, ve merakmn bilgisayarlar diyarna ulamas uzun sürmedi. Önce arkadaşarımın bilgisayarların kartırdm, onlar bozdum, ve bol bol azar iittim. Böylece bozduum eyleri düzeltmeyi öğrendim.

İlkokul son snfta ailem nihayet bana bir bilgisayar almaya karar verdi. nter-nete eriebilmem daha uzun sürdü. Bu arada, bilgisayar dergilerini okuyup (O zamanlar daha dolulard) CDlerindeki programlar denedim. Bir sefer CDler-den biri Vector Linux diye bir ey veriyordu, fakat “iletim sistemi” kavramn pek anlamadmdan ilgimi çekmedi, bir kenara atp unuttum. Komik bir ekilde, çok daha sonra, bakaların da çok benzer aamalardan geçtiini gördüm: <http://buraxezer.wordpress.com/2007/07/20/linux-maceram-nasil-basladi/>

Bir yerde BASIC ile oyun programlama ile ilgili bir kitap buldum ve programlama ile ilgilenmeye baladm. Bilgisayara kendi istediimi yaptrabilmek çok houma gidiyordu, bilgisayarların amac bu deil miydi zaten? Daha sonra Visual Basic ile tantm (evet, evet, acemiymdim!), arkadaşarım sinir etmeye ve matematik ödevlerimi çok uramadan yapmaya yönelik programlar yazp disketler aracıyla arkadaşarına dattm. Nesnelerle programlamay (OOP) anladm sandm (sonuna nokta koyunca o nesnenin altındaki eylere ulalıyordu ite, bu kadar basit!).

Evime internet ilk balandnda hayatm deiti diyebilirim. Artk snrl saydak kitaplara ve programlara baml deildim, internette bir konudaki bir kayna been-mezsem bir dierini bulmam bir dakika bile sürmüyordu. Büyük bir bollukla karlatm ve hepsini kucakladm. Sonra telefon faturası geldi, ve zamanm biraz daha dikkatli kullanmay öğrendim.

nternetin bana faydalar dokundu: bir sürü baka programlama dili de varm, onu öğrendim. İkkin kendime geocities’de web sitesi açtm ve arkadaşarına hava attm. Fakat tklayarak site hazırlamak pek programlama sayılmazd, ben “Kayna Görüntüle” kısmında yer alanlara daha meraklydm. Buradakileri deitirerek bakaların sitelerini deitirebileceimi sanarak kendi kendime hnzr kakhahalar attm, olmadn görünce biraz bozuldum. Yine de HTML öğrenmeye karar verdim ve öğrenirken deikenlerin, döngülerin ve benzer eylerin olmadn görüp bunun sadece bir tanım-lama dili olduunu fark ettim. Bunun üzerine belli girdilere göre farklı HTML çkıtlar çıkarmakta uzmanlaan programlama dillerine merak saldm ve özellikle

PHP ile ilgilenmeye baladm. IRC ile tantm.

Bu arada, “internetteki insanlar” açık kaynak diye bir ey konusunda çok heyecanlıydılar. Bakaların programları okuyabilmek ve deitirebilmek mi? Bence pek ho bir fikirdi. Kimi insanların bu fikre neden bu kadar kar olduunu anlayamadım, çünkü o zamanlar parasal çıkarlar, tekelcilik gibi konular konusunda pek bir fikrim yoktu. Eric S. Raymond’un ve benzerlerinin yazları okuyup ksmen anladım, Jargon File’ bol bol karırdım, bak açm geniledi. LISP diye bir dilden söz edildi, fakat örnek programlara bakınca pek anlayamadım, unuttum. letim sistemi kelimesini tekrar duyar oldum, ve bu sefer açık kaynak kavram ile birleerek karma Linux’u çıkardı. Dosyaların (ilk CD yazcmla) yedekledim ve en yakın arkadaşmdan aldım Red Hat Linux CDsini kurmaya karar verdim. Kurulum baarl geçti fakat aynı sırada LILO, MBR’da sadece kendine yer olduuna karar verdi (suçlu ben deildim, vallahi!) ve ben uzunca bir süre Linux’la babaa kaldım, konsolla tantm. Houma gitti.

rili ufaklı bir sürü baka programlama diline bulatm, arkadaşların Liselere Giri Snav’na çalıp dershaneye giderken ben bir programlama kursuna gittim, sanırım bolca yarar oldu. Programlama merakım peini brakmamam saladılar. Programlama mantım geliti, birçok eyin dorusunu öğrendim, fakat çok sonradan fark ettim ki her eyin deil. Aynı okulun lisesine devam ettim ve pıman da olmadım. Lise boyunca zamanım yettiince programlamaya devam ettim. Pardus ile ve Freenode Pardus kanalındaki insanlar ile tantm, Pardus’u kullandım, projeye az da olsa katkıda bulunmaya çaltım.

ÖSS denemelerinde kötü deildim, fakat istediim üniversitelere girmeyi garantileyecek kadar da baarl olmadım. Ders durumum ve ingilizcem iyiydi, bu yüzden yurtdnda okumay kafama koydum ve yurtdna bavurmaya baladm. Bilgisayar bölümü iyi olan okullara bavurmaya dikkat ettim. Kabul edildiim okullar arasından Worcester Polytechnic Institute’u (<http://www.wpi.edu/>) seçerek hayatmda yeni bir serüvene adm attım.

WPI’da ilk senem boyunca Scheme (evet, bir LISP türevidir!) araçları ile fonksiyonel programlama ve lambda’nın gücü, Java araçları ile nesnel programlama, önce C ve daha sonra C++ ile sistem programlama ve veri yapıları (listeler, kuyruklar, ağaçlar? Sanırım bir teknik sözlüğe ihtiyacım var), sayısal devreler, basit mikroilemlerin çalışma prensipleri, az da olsa assembly ve son olarak da algoritma teorisi konularında bilgi edindim. Eski yazdıklarıma nazaran daha büyük ve karmaşık programlar yazmaya altım. u an ilk senemi bitirmek üzereyim, ve sonunda istediim eyleri öğrenmekte olduuma mutluyum.

Çok ksa bir ekilde bilgisayar dndaki hayatmdan bahsetmek gerekirse, hava kouları izin verdii sürece yelken yaparım veya yüzerim. lginç olmas kouluyla bolca kitap okurum, eski kitapları karırmay severim. Yaklaşık bir buçuk sene önce kendi kendime gitar çalmayı öğrenmeye baladm, her frsatta oturup biraz çalarım. Kafam dinlemeyi severim. Bazen panik yapabilirim fakat sinirlendiim pek görülmemiştir.

### 3 Neler yapabilirim?

Bir önceki bölüm buna biraz k tutsa da, yapabildiklerim bunlarla snrl deil. te bir liste:

- Kendime güvenecek kadar çok tecrübeli olmadm türlü türlü ilginç dili bir kenara braksak, PHP, C ve C++, Python, Scheme ve Java ile u an rahatça program yazabilirim.
- SQL'i, ve dolaysyla MySQL, postgresql ve sqlite gibi veritaban sistemlerini rahatlıkla kullanabilirim.
- PHP tecrübem sayesinde web programlama konusunda iyiyim, kullanıc ve sunucu arasında veri iletiiminin nasıl saland konusunda bilgim var. Herkesin çok heyecanland AJAX denilen şeyin aslında XMLHttpRequest nesnesinden ibaret olduğunu, benzer bir anlamsız heyecanın geçmişte Javascript, CSS ve DOM'un birleşimine DHTML denmesiyle olutunu da biliyorum. Hayır, hiç Django kullanmadım, fakat Codeigniter (<http://codeigniter.com/>) gibi MVC sistemleriyle tecrübem var, ve Django da korkunç gözüküyor. Bunun dnda site tasarımı yaparken neden tabloların tasarımı amacıyla kullanılmaması gerektiğini, nasıl CSS kullanılarak veri ile sunumun katmanlarının ayrılabilirliği biliyorum. W3C standartlarına uygun sayfalar hazırlayabilirim.
- Linux ile har neirim.
- “yi” programlama anlayışının ne olduğunu, “temiz” kod derken neyin kastedildiğini, dokümantasyonun önemli olduğunu ve yazarken biraz şıksam da sonradan herkesi gereksiz zaman harcamaktan koruyacağım ve çok mutlu edeceğimi, büyük bir projeye başlamadan planlamam bana sonradan efor kazandıracan biliyorum.
- Zorluk çekmeden teknik İngilizce kullanabilir ve İngilizce rapor yazabilirim.
- ahsı olarak: Detaylara meraklıyım, fakat pratik olmay da gözden çıkarmam. Yeni şeyler öğrenmekten çekinmem, bu nedenle de yapıcı eleştiriyi her zaman açımdır. Oturup düşünerek ve / veya araştırarak ilginç bir soruna çözüm getirmeye çalışmak hoşuma gider, hatta beni geceleri ayakta tutar.

### 4 Hangi projelere katkıda bulunabilirim?

İlgili olduğum, yarar sağlayabileceğimi düşündüğüm birden fazla proje var. Sizlere kolaylık olması için ve kendimi tekrarlamamak adına bunlar bölümlere ayırdım. Her bir projeden bahsetmediysem, ilgili olmadımdan değil, konu(lar) / yazıl(m)lar üzerinde yeterince tecrübem olmadı ya da kısa zamanda yeterli tecrübe edinemeyeceğimi düşündüğümüdür. Her siz benimle aynı fikirde değilseniz lütfen fikrimi deitirmekten çekinmeyin!

## Web tabanlı projeler

Web programlama ve web arayüzü hazırlamada tecrübeli olduüm düşünülürse bu tam bana uygun bir projeye benziyor. Dediim gibi, PHP ile web programlamada rahatım fakat Python tercih ettiinizi tahmin ediyorum. Django ile ilgili tecrübem yok, fakat sitedeki belge saysna bakınca ve benzer framework'leri kullandım düşünürsek öğrenmek çok da büyük bir sknt olmasa gerek. Ayrıca veritabanı temelli web siteleri olutmaya da alkn olduüm düşünülürse (veritabanı temelli olmayan u dönemlerde az zaten), sanırım gereklilikleri tamamlam oluyorum. Tasarımın kim tarafından yapılcan bilemiyorum, fakat eer grafik tasarımıncı tarafından yapılacaksa da grafik tasarım programlarının yarattıkları karmak ve kimi zaman saçma html çıktıların temiz, standartlara uygun ve birçok tarayıcıda çalışır hale getirmekle de bir sorunum yok. Bu kategoride yarar salayacağım düşündüğüm iki proje var:

### Pardus Yeni Fikirler Web Arayüzü

Ubuntu Brainstorm projesinin dnda Dell'in de Ideastorm (<http://www.ideastorm.com/>) diye benzer bir sitesi var, fakat zamanında Linux ile gelen bilgisayar isteklerini sansürlemeye çalışınca olay çıkmı. Sonuç: Dell artık Ubuntu yüklü bilgisayarlar satıyor. Bundan bahsetmem böyle bir sansür eyleminin olacan düşündüğümde deil. Sadece, böyle bir site ortaya çıkınca kullanıcı kitlesinin genel düşüncesi her istediklerinin olması gerektiği kansında oluyor. Yani gün gelir de kullanıcılar projenin misyonuna, (ya da Dell'in durumunda çıkarsa) kart olan ya da projenin her ne nedenle olsun kalkmak istemeyeceği bir fikir ortaya çıkarsa, bu isteklerinin geliştiriciler tarafından uygulanacağını kesin olmad da belirtilmeli. Tabii ki kendi bana bu isteklerini yerine getirmelerini engelleyen yok, açık kaynak açık kaynaklıdır.

### Benim Pardus'um Web Arayüzü

Bunun gibi bir eyni neden daha önce hazırlanmadı merak ediyordum! Kimi datm kurulum CDlerini modüler hale getiriyor, böylece tecrübeli kullanıcılar bir dizindeki dosyaları deitirerek yüklenecek paketleri deitiriyorlar. Sayı sayıda datm da kullanıcılara bir arayüz sağlıyor (mesela Slax, modüllerle özellikle önem veren bir minyatür datm, bolca örnek alınacak özelliği var). Fakat farklı kullanıcıların kendi kombinasyonlarını paylaşımlarını salayacak böyle bir platformun daha yapıldı sanmıyorum, dolayısıyla ses getirmesi olası. Kullanıcıların birbirlerinin proje dosyalarına oy vermeleri, birbirlerine öneride bulunmalar ve eileti yapmalar salanabilir.

Biraz daha üretken olup kullanımı kolay göz önüne alınması düşünülürse siteye bir XML web servisi veya benzeri bir arayüz oluturularak kullanıcıların site içeriine tarayıcı açmadan, sihirbaz içinden ulaşabilmesi çok hoş bir ek özellik olabilir, fakat bu iki takımın beraber çalışması da gerektirecektir.

## Masaüstüne likin Projeler

Python, C, C++ ve Java'nın her birini kullanılmı var. DBUS' kullanma ve GUI oluturma konusundaki tecrübem ise ne yazık ki sınırlı, fakat öğrenebileceklerimin sınırı yok! Pisi paketleri hakkında bilgime gelince, daha önce paketlerin içeriğini açıp karıştırdım, pisi belgesini de okudum, fakat hazırlayıp da repoya baladım paketim, ve dolayısıyla pratik tecrübem yok.

## Pardus CD/DVD/USB İmaj Hazırlama Sihirbaz

Kullanıcılar bakımından çok hoş olsa gerek, çünkü kullanıcıların çoğu kendilerine hitap eden eylemlere bayılır. Kamıyca Facebook ve Myspace gibi sitelerin rakiplerini amallarının büyük bir nedeni budur; Myspace'te kişiselleştirme o kadar sapsık bir halde ki 1998 civarlarının çığıcı <bgsound>'lu, <blink>'li, Javascript'i bol bol gereksiz yere sömüren sitelerini aratmıyor. Facebook'da durum çok daha sakin olsa da aynı konumu “kurbana olsanız ne tür vraklardanız” ekinde testler ve benzeri bir sürü uygulama durumu ele geçirmi duruyor. Bunun dında Pardus'u kullanmak isteyen şirketler ve makinelerini satarken Pardus'u kurulu olarak kullanıcılarına sunan üreticiler markalama yapmak isteyebilirler, bunu kolay bir ekinde salamak için bir seçenek oluturulabilir.

Tabii ki sırf ego tabanlı da deil. Ciddi ihtiyaçlar olanlar, mesela birçok makineye standart kurulum yapmak isteyenler de olacaktır. Bunlar için YALI'y grafik arayüzü çalıştırmadan kullanan (böyle bir ey var mı?) ve önceden belirlenmiş bir takım paketi kuran kurulum imajlar oluturulabilir. Kısık alana sahip olanlar için “taban” sistemi kurma opsiyonu olması düşünölmütür zaten, tahminimce. Bunun da dında, kaynak kodunun çok iletim sistemi baml olmadan yazılmasına dikkat edilirse Debian'n Jigdo'sunun (<http://atterer.net/jigdo/>) yapt gibi bir Windows sürümü de çıkartılarak bu platformdan geçi yapmaya çalışan (ya da yapmaya meyyal) kullanıcılara klavuz olunabilir. Fakat bu bal bana bir proje haline de dönüebilir, ayrıca takımı bu konuda görüü ne olur, bilemem. Buna alternatif olarak sadece imaj oluturmaya yönelik çok hafif bir çalışan CD de olabilir. İmaj olutururken kullanıcıya kaynak olarak repolar yanında makinedeki bir dizinin kullanılmasına da izin verilmesi hoş olacaktır, böylece paketleri tekrardan indirme zahmetine katlanılmak zorunda kalınmaz. Son olarak, programın çalışan CD türü imajlar üretebilmesi de kimi kullanıcıların houna gidebilir.

## Bildiri Yöneticisi

Anabirim tasarım prensipleri bakımından çok mantıklı, kullanılan birkaç farklı programın birbirinden habersiz olarak aynı köden bildiri çıkarmas rastlanmaz bir durum deil, ve çirkin. Üstelik amaç kullanıcıya tutarlı bir arayüz sunmaksa, daha da mantıklı. Ayrıca uyar seviyelerinin tanımlanabilmesi özelliği eklenirse “u an iim var (ya da openarena oynuyorum!), kritik uyarlar dında beni rahatsız etme” veya “kritik seviyedeki bildirileri e-mailime gönder” ekinde seçenekler verilebilmesi salanabilir. Son olarak, geliştiriciler için DBUS etrafına bir wrapper API (yada birden fazla dil için APIler) oluturulabilir ve böylece geliştiricilerin DBUS ile didik didik uramadan bildiri yöneticisini kullanabilmeleri salanır olur.

## **Pardus Hata Ayklama**

Farkl sorunlara kafa yorarak çözüm getirmek houma gittiinden dolay, ve merakl olmam nedeniyle birçok ey hakknda az ya da çok bilgim oldu için bu projeye de katkıda bulunabileceime inanyorum. Ayrıca bu projede her an birçok insanla iletiim halinde bulunma ansna sahibim, ki bu da çok ho.

## **5 Ne zaman?**

Ne zaman çalabileceim konusuna gelirsek, Mays'n 15'inden Haziran sonu kaba bir tahmin olabilir. Fakat bu tarihlerle oynanabilir, tarihler kaydrılabilir. Mays banda stanbul'a geliyorum, Austos civar da okuluma geri dönüyorum. Çalırken Gebze'de mi olmam gerekiyor ya da evden çalmam m bekleniyor, bilirim, bir süre belirtmem de kolaylaabilir. Lütfen benimle irtibata geçmekten çekinmeyin.

## **6 Kaynak Kod Örnekleri**

Bu aamaya kadar geldiyseniz size teekkür ediyor, ilginize mutlu oluyor, sabrınız takdir ediyorum. Sizi daha çok da fazla yormadan houma giden (ve harddiskimi kartırırken bulduma atm) birkaç kod parçacın size sunuyorum:

## Scheme Denemeleri

Kendi kendime deneyip ilginç buldum örnekler. Neredeyse hiç deer atanmamasna, recursion dnda döngü kullanılmamasna, fonksiyonların deer olarak kullanılabilmesine dikkatinizi çekerim.

```
;;;;;;;;;;;;; Treemap ;;;;;;;;;;;;;;

;A treenode is (make-treenode num treenode treenode)
(define-struct treenode (val left right))

;sample tree:
(define tree1
  (make-treenode 7
    (make-treenode 4
      (make-treenode 2 empty empty)
      (make-treenode 8 empty empty)
    )
  ))

;treemap: treenode f(x) -> treenode
;applies f to whole tree
(define (treemap node func)
  (cond
    [(empty? node) empty]
    [(treenode? node)
     (make-treenode
      (func (treenode-val node))
      (treemap (treenode-left node) func)
      (treemap (treenode-right node) func))]
  ))

(treemap tree1 (lambda (x) (* x x))) ;square all nodes

;;;;;;;;;;;;; List search ;;;;;;;;;;;;;;

;num-exists: num list[num] -> boolean
;does the num exist in the list?
(define (num-exists num alon)
  (ormap
    (lambda (x) (= x num))
    alon)
  )
(num-exists 7 '(4 2 3 7 5 1)) ;true

;;;;;;;;;;;;; Quicksort ;;;;;;;;;;;;;;

;smaller: num list[num] -> list[num]
```



```

;return elems of list smaller than num
(define (smaller num list)
  (filter
    (lambda (x) (> num x))
    list))

;larger num list[num] -> list[num]
;return elems of list larger than num
(define (larger num list)
  (filter
    (lambda (x) (< num x))
    list))

;qsort list[num] -> list[num]
;sort the list using quicksort
(define (qsort lst)
  (cond [(empty? lst) empty]
        [(cons? lst)
         (append
          (qsort (smaller (first lst) (rest lst))) ;smaller
          (list (first lst)) ;center
          (qsort (larger (first lst) (rest lst))) ;larger
          )
         ]))

(qsort '(7 5 4 6 9 3)) ; (3 4 5 6 7 9)

```

## Pointer'lar ve C içinde Recursion

Sistem programlama dersimdeki laboratuvar çalmalarından biri. Amaç öğrenci verileri içeren bir string girdi alıp, stringi parse edip, öğrencileri numara sıralarına göre treeye yerletirmektir. Ben de bir süre önce aldım fonksiyonel programlama dersinin etkisi altında kalınca öğrenci ekleme ve çıktı fonksiyonların recursion ile yazmaya karar verdim. Bol bol pointer cambazı da var.

```
//recursively insert new student to tree
void insert(Node **n, int studentid, int g1, int g2, int g3){
    if (!(*n)){ //if null, insert data
        (*n) = malloc(sizeof(Node));
        (*n)->studentid = studentid;
        (*n)->g1 = g1;
        (*n)->g2 = g2;
        (*n)->g3 = g3;
        (*n)->left = NULL;
        (*n)->right = NULL;
        return;
    }
    if (studentid < (*n)->studentid){ //if less go left
        insert(&(*n)->left, studentid, g1, g2, g3);
    } else {
        insert(&(*n)->right, studentid, g1, g2, g3);
    }
}

//recursively print tree
//starts from leftmost (smallest) node and goes right
void printTree(Node **n){
    if((*n)->left){
        printTree(&(*n)->left);
    }
    printf("ID: %d Grades: %4.2f %4.2f %4.2f\n",
        (*n)->studentid, (*n)->g1, (*n)->g2, (*n)->g3);
    if((*n)->right){
        printTree(&(*n)->right);
    }
}
```

## İlk Makine Kodu Programı

Bu, makine organizasyonu dersimde yazdığım ilk makine kodu programıdır:

```
;----- ../:Main Program::\.. -----
;INIT - set values
0110 100 010 000000      ;R4 = data at the address in R2
0101 110 110 1 00000     ;clear test register
0101 111 111 1 00000     ;clear counter
;BEGINNING - start of main loop
0001 100 100 1 00000     ;set the cond. codes for our bit pattern
0000 1 0 0 000000010     ;branch to ONE part if high bit is 1
;ZERO - if high bit is 0
0001 100 100 000 100     ;left shift bit pattern
0000 1 1 1 000000011     ;branch out to the FINAL part
;ONE - if high bit is 1
0001 100 100 000 100     ;left shift bit pattern
0001 100 100 1 00001     ;add 1 to the end
0000 1 1 1 000000000     ;branch out to the FINAL part
;FINAL - last part of the loop
0001 111 111 1 00001     ;increment loop counter
0001 110 111 111000      ;subtract 8 from r6 so that n flag is set
                                ;while 8 loops haven't been done
0000 1 0 0 11110110     ;branch back to BEGINNING if n
1111 0000 00100101      ;halt
;END
```

Bunun assembly karlı şöyle:

```
;initialize registers
        LDR      R4, R2, 0      ;R4 = data at the address in R2
        AND      R6, R6, #0     ;test = 0
        AND      R7, R7, #0     ;counter = 0

;begin main loop
BEGIN    ADD      R4, R4, #0     ;set the cond codes for val. in R4
        BRn      ONE           ;if 15th bit is 1, jump to ONE

ZERO     ADD      R4, R4, R4     ;left shift
        BR       FINALLY       ;all done, jump to finally

ONE      ADD      R4, R4, R4     ;left shift
        ADD      R4, R4, #1     ;add 1
        BR       FINALLY       ;all done, jump to finally

FINALLY  ADD      R7, R7, #1     ;increment counter
        ADD      R6, R7, #-8    ;subtract 7 for BR
        BRn      BEGIN         ;return to the beginning if c < 7
```

HALT  
.END

in komik taraf, operasyon