

Termcap Kütüphanesi ve Veritabanı

Yazan: **Richard M. Stallman** Çeviren: **Nilgün B. Bugüner**

Nisan 2006

Özet

Termcap uygulamaların uçbirimlerin yeteneklerini kullanabilmelerini olanaklı kılan bir veritabanı ve bir kütüphanedir. Bir uçbirim için kaydedilmiş bilgilerden örnek vermek gerekirse, uçbirimin kaç sütun genişlikte olduğu, imlecin belli bir konuma taşınıp dizgenin oraya gönderilmesinin olanaklı olup olmadığı (satır ve sütun numaralarının nasıl kodlanacağı), ekran içeriğinin yukarı ve aşağı kaydırılabilirliği ve bu işlemin kaçar satırlık yapılabileceği gibi bilgilerin varlığından bahsedebiliriz. Termcap kütüphanesi ise ekranı uçbirimden bağımsız olarak kullanmak isteyen karakter temelli uygulamaların bu veritabanına kolayca erişebilmesini olanaklı kılmak için tasarlanmıştır.

Bu kitap, 7 Kasım 1998'de güncellenen [ikinci baskının^{\(B1\)}](#) çevirisidir.

İçindekiler

1. Giriş	4
2. Termcap Kütüphanesi	4
2.1. Termcap Kütüphanesinin Kullanıma Hazırlanması	4
2.2. Uçbirim Tanımının Bulunması	5
2.3. Uçbirim Tanımının Sorgulanması	6
2.4. Termcap Kullanımı için İklendirme	8
2.5. Geciktirme	8
2.5.1. Geciktirme Niçin Gerekir ve Nasıl Yapılır	8
2.5.2. Geciktirme Yetersiz Kalırsa	9
2.5.3. Uçbirim Tanımında Geciktirmenin Belirtilmesi	9
2.5.4. tputs ile Geciktirme	10
2.6. Parametreler	11
2.6.1. Kodlamanın Açıklaması	11
2.6.2. Ekran Komutlarının Parametrelerle Gönderilmesi	14
3. Veritabanının Biçimi	15
3.1. Uçbirim Tanımının Biçimi	15
3.2. Yeteneklerin Yazılması	16
3.3. Uçbirim Türü İsimlendirme Uzlaşmaları	17
3.4. İlgili Tanımları Miras Alma	18
3.5. Etkin Veritabanı Değişirse	19
4. Uçbirim Yeteneklerinin Tanımları	19
4.1. Temel Karakteristikler	19

4.2. Ekran Boyutları	21
4.3. İmleç Hareketleri	21
4.4. Sarmalama	25
4.5. Kaydırma	25
4.6. Pencereler	27
4.7. Ekran Parçalarının Temizlenmesi	28
4.8. Satır Ekleme/Silme	28
4.9. Karakter Ekleme/Silme	29
4.10. Çarpıcı ve Sıradan Görünüm Kipleri	33
4.11. Altçizme	35
4.12. İmleç Görünürlüğü	36
4.13. Çan	37
4.14. Tuştakımı ve İşlev Tuşları	37
4.15. Meta Tuşu	39
4.16. İklendirme	39
4.17. Geciktirme Yetenekleri	40
4.18. Durum Satırı	41
4.19. Yarım Satırlık Hareket	42
4.20. Uçbirimlerle İlintili Yazıcıların Denetimi	42
5. Yetenek İsimlerinin Listesi	43
5.1. Seçimlik Yetenekler	43
5.2. Sayısal Yetenekler	45
5.3. Dizgesel Yetenekler	45
Değişkenler ve İşlevler Dizini	53
Yetenekler Dizini	54
Kavramlar Dizini	57

Legal Notice

Copyright © 1988 Free Software Foundation, Inc.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.

Yasal Uyarı

Telif Hakkı © 1988 Free Software Foundation, Inc.

Bu kılavuzun harfi harfine kopyalanmasına ve dağıtılmasına telif hakkı uyarısının ve bu izin uyarısının tüm kopyalarında bulunması şartıyla izin verilmiştir.

Bu kılavuzun değiştirilmiş kopyalarının kopyalanması ve dağıtılması, türetilmiş çalışmanın sonuçlarının tamamı, harfi harfine kopyalamanın koşullarını içererek ve buna eşdeğer bir izin uyarısı da sağlandığı takdirde izin verilmiştir.

Bu kılavuzun diğer dillere çevirilerine, bu izin uyarısını bulunduğu Free Software Foundation tarafından onaylanmış bir çeviri değilse, yukarıda açıklanan değiştirilmiş sürümlerinin koşulları sağlandığı takdirde izin verilmiştir.

Feragatname

BU KİTAP ÜCRETSİZ OLARAK RUHSATLANDIĞI İÇİN, İÇERDİĞİ BİLGİLER İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI OLARAK BELİRTİLMEDİĞİ MÜDDETÇE TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR KİTABI "OLDUĞU GİBİ", AŞIKAR VEYA ZIMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BİLGİNİN KALİTESİ İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATALI BİLGİDEN DOLAYI DOĞABİLECEK OLAN BÜTÜN SERVİS, TAMİR VEYA DÜZELTME MASRAFLARI SİZE AİTTİR.

İLGİLİ KANUNUN İCBAR ETTİĞİ DURUMLAR VEYA YAZILI ANLAŞMA HARİCİNDE HERHANGİ BİR ŞEKİLDE TELİF HAKKI SAHİBİ VEYA YUKARIDA İZİN VERİLDİĞİ ŞEKİLDE KİTABI DEĞİŞTİREN VEYA YENİDEN DAĞITAN HERHANGİ BİR KİŞİ, BİLGİNİN KULLANIMI VEYA KULLANILAMAMASI (VEYA VERİ KAYBI OLUŞMASI, VERİNİN YANLIŞ HALE GELMESİ, SİZİN VEYA ÜÇÜNCÜ ŞAHISLARIN ZARARA UĞRAMASI VEYA BİLGİLERİN BAŞKA BİLGİLERLE UYUMSUZ OLMASI) YÜZÜNDEN OLUŞAN GENEL, ÖZEL, DOĞRUDAN YA DA DOLAYLI HERHANGİ BİR ZARARDAN, BÖYLE BİR TAZMİNAT TALEBİ TELİF HAKKI SAHİBİ VEYA İLGİLİ KİŞİYE BİLDİRİLMİŞ OLSA DAHI, SORUMLU DEĞİLDİR.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim, bir ticari isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.

1. Giriş

Termcap uygulamaların uçbirimlerin yeteneklerini kullanabilmelerini olanaklı kılan bir veritabanı ve bir kütüphanedir. Berkeley Unix'in uçbirim kütüphanesi temel alınmıştır.

Termcap veritabanı yüzlerce farklı uçbirimin yeteneklerini oldukça ayrıntılı bir biçimde içerir. Bir uçbirim için kaydedilmiş bilgilerden örnek vermek gerekirse, uçbirimin kaç sütun genişlikte olduğu, imlecin belli bir konuma taşınıp dizgenin oraya gönderilmesinin olanaklı olup olmadığı (sıra ve sütun numaralarının nasıl kodlanacağı), ekran içeriğinin yukarı ve aşağı kaydırılabilirliği ve bu işlemin kaçar satırlık yapılabileceği gibi bilgilerin varlığından bahsedebiliriz.

Termcap kütüphanesi, ekranı uçbirimden bağımsız olarak kullanmak isteyen karakter temelli uygulamaların bu veritabanına kolayca erişebilmesini olanaklı kılmak için tasarlanmıştır.

Bu kılavuzda, termcap kütüphanesinin Unix sürümüne göre bazı ek oluşumlar içeren GNU sürümü anlatılmıştır. Ek oluşumların neler olduğu ve dolayısıyla Unix termcap'ının nasıl kullanılabileceğinden ayrıca bahsedilmiştir.

Özgür yazılımlarda kullanmak için, termcap kütüphanesinin GNU sürümü özgür kaynak kodlu olarak dağıtılmaktadır. Unix ve VMS sistemlerinde (en azından) çalışmaktadır. Dosyaların GNU Emacs ile ilgili olanları `termcap.c` ve `tparam.c` dosyalarıdır.

Bu kılavuz, kullanıcı uygulamaları için Unix ile temelde uyumlu tamamen özgür bir işletim sistemi geliştirilmesi amacıyla GNU projesi için yazıldı. Projenin yaklaşık olarak üçte ikisi tamamlandı. GNU Emacs metin düzenleyici ve oldukça taşınabilir olarak eniyelenmiş C derleyicisi de dahil olmak üzere, GNU projesi hakkında daha ayrıntılı bilgi edinmek istiyorsanız aşağıdaki adrese bir dolar göndermeniz yeterlidir (Belgenin özgün sürümündeki adres yerine bu çeviriye şu an geçerli olan adres yazılmıştır. Bu adres merkez büro adresidir.):

Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor,
Boston, MA 02110-1301 USA

2. Termcap Kütüphanesi

Termcap kütüphanesi uygulama geliştiricilerinin termcap veritabanına erişimin kolaylaştıran bir arayüzdür. Aşağıda sıralanmış amaçlara uygun işlevler içerir:

- Kullanıcı uçbirimi türünün tanımının bulunması (`tgetent` işlevi).
- Çeşitli konularda bilgi edinmek için veritabanının sorgulanması (`tgetnum`, `tgetflag`, `tgetstr` işlevleri).
- Gecikmelerin hesaplanması ve uygulanması (`tputs` işlevi).
- Uçbirime özgü imleç konumları gibi sayısal parametrelerin ekran komutlarının gerektirdiği biçimde kodlanması (`tparam`, `tgoto` işlevleri).

2.1. Termcap Kütüphanesinin Kullanıma Hazırlanması

Termcap kütüphanesini yazılımınızda kullanmak istiyorsanız iki işlem yapacaksınız:

- Derleyicinin gereksinimlerini karşılamak için kütüphanedeki işlevlerin ve değişkenlerin bildirimleri.

GNU sistemlerinde bu işlem, bu değişken ve işlevlerin kullanıldığı her kaynak dosyasının `termcap.h` başlık dosyasını içermesi sağlanarak yapılır.

Unix sistemlerinde ise çoğunlukla böyle bir başlık dosyası yoktur. Bu nedenle değişkenleri `external` olarak kendiniz bildirmelisiniz. Bunu işlevler için de yapabileceğiniz gibi, isterseniz, örtük olarak bildirir ve değerlerine de ilgili türden `int` türüne dönüşüm uygularsınız.

Her termcap kütüphane işlevini argümanlarının nasıl aktarılacağını belirtmek amacıyla ANSI C prototipleriyle bildirim olarak göstereceğiz. Eğer GNU C derleyicisi kullanmıyorsanız, büyük ihtimalle işlev prototiplerine ihtiyacınız olmayacağından bildirimlerinizde argüman isimlerini ve türlerini yoksayabilirsiniz.

- İlintileyici (ld) kütüphaneyi bulmaya çalışacaktır. Kütüphaneyi bulmasını sağlamak için yazılımınızı derlerken seçenek olarak **-ltermcap** veya **-ltermcap** belirteceksiniz.

2.2. Uçbirim Tanımının Bulunması

Bir uygulama yazılımı termcap'ı kullanacaksa önce kullandığı uçbirim tanımına bakmalıdır. Bu işlem **tgetent** işlevi çağrılarak yapılır.

```
int tgetent (char *tampon,                                     işlev
             char *uçbirim_türü);
```

Bu işlev tanımı bulur ve hatırlar; böylece ilgilendiğiniz [uçbirim yeteneklerini sorgulayabilirsiniz](#) (sayfa: 6).

uçbirim_türü aranacak uçbirim türünün ismini içeren bir dizge olmalıdır. Bu değeri bir **getenv ("TERM")** çağrısı ile **TERM** ortam değişkeninden elde edebilirsiniz.

Termcap'ın GNU sürümünü kullanıyorsanız, **tgetent**'ten gerekli yeri kendisinin ayırmasını isteyebilirsiniz. İşleve **tampon** olarak bir boş gösterici aktarırsanız, **tgetent malloc** işlevini kullanarak gerektiği kadar yeri kendisi ayıracaktır. Ayrıca, ayrılan bellek alanının adresini almanın bir yolu olmadığından, bu bellek alanını serbest bırakma çabasına da girmeyeceksiniz.

Termcap'ın Unix sürümünü kullanıyorsanız, tanımı içerecek alanı kendiniz ayırmalı ve adresini **tampon** argümanı ile işleve aktarmalısınız. Ne kadar alan gerektiğini söylemek mümkün olmadığından, teamüllere uygun olarak 2048 baytlık bir tampon yeterli olacaktır diyebiliriz. (Önceleri, uzlaşım 1024 baytın yeterli olacağı şeklindeydi, ama günlerden bir gün bir uçbirim türü için bu yetersiz oldu.)

Elde edilen tanımın nasıl saklandığı bu kılavuzun konusu değildir. Termcap bu tanımı kendi saklar ve bu tanımı **tgetnum**, **tgetstr** veya **tgetflag** çağrılarınıza yanıt vermek için kullanır. Eğer tamponu termcap ayırmışsa, **tgetent**'i tekrar çağırdığınızda termcap bu belleği serbest bırakır. Eğer tamponu siz ayırmışsanız, tanımın sorgulanması gerektiği sürece bu tamponun içeriğinin değişmeden kalmasını sağlamalısınız.

Uçbirim türünün veritabanına erişimde bir zorlukla karşılaşmışsa, **tgetent -1** değeriyle döner. Eğer, veritabanına sorunsuz erişildiği halde belirtilen uçbirim türü veritabanında yoksa 0 ile döner. Diğer durumlarda bunlardan farklı bir değerle döner.

Burada **tgetent** işlevini nasıl kullanmanız gerektiği gösterilmiştir:

```
#ifdef unix
static char term_buffer[2048];
#else
#define term_buffer 0
#endif

init_terminal_data ()
{
    char *termtype = getenv ("TERM");
    int success;

    if (termtype == 0)
```

```

hayati ("`setenv TERM <uçbirimtürü>' "
        " komutu ile bir uçbirim belirtmelisiniz.\n");

success = tgetent (term_buffer, termtype);
if (success < 0)
    hayati ("Termcap veritabanına erişilemedi.\n");
if (success == 0)
    hayati ("`%s' diye bir uçbirim tanımlanmamış.\n", termtype);
}

```

Buradaki **hayati** işlevinin bir hata iletisi basıp uygulamayı sonlandıran bir işlev olduğu varsayılmıştır.

TERMCAP ortam değişkeni tanımlıysa, değeri uçbirim türü veritabanını hükümsüz bırakmakta kullanılır. **tgetent** işlevi **TERMCAP** değerini özdevinimli sınar. Eğer isim bir / ile başlıyorsa, değer, standart veritabanı olan **/etc/termcap** yerine veritabanı olarak kullanılacak dosya ismi olarak ele alınır. Eğer değer bir / ile başlamıyorsa, uygulanması istenen **uçbirim_türü** uçbirim türünün tanımı için kullanılır. Uçbirim tanımı ile ilgili daha ayrıntılı bilgi edinmek için [Veritabanının Biçimi](#) (sayfa: 15) bölümüne bakınız.

2.3. Uçbirim Tanımının Sorgulanması

Bir uçbirim tanımında kayıtlı her bilgi parçasına **yetenek** adı verilir. Kayıtlı her uçbirim yeteneğinin özel bir anlamı olan ve iki harfle kodlanan bir ismi vardır. Örneğin sütun sayısının ismi **co**'dur. Standart yetenek isimlerinin tamamını [Uçbirim Yeteneklerinin Tanımları](#) (sayfa: 19) bölümünde bulabilirsiniz.

tgetent ile [uygun uçbirim tanımını bulduktan](#) (sayfa: 5) sonra, artık uygulama yazılımınız çeşitli uçbirim yeteneklerini **sorgulamaya** hazır demektir. Aradığınız yetenek ile ilgili bilgi almak için yeteneğin iki harflik kodunu belirtmeniz gerekir.

Yeteneklerin değerleri sayısal, mantıksal (yetenek var/yok tarzında) ya da dizge biçiminde olabilir. Her yetenek değerinin kendine özgü bir değer türü vardır. Örneğin, **co** daima bir sayısal değer içerirken **am** (kenarlarda özdevinimli sarmalama) daima bir var/yok değeri içerir. **cm** (imleç hareket komutu) ise daima bir dizge değere sahiptir. Her yeteneğin değerinin hangi türde olduğu bu kılavuzda açıklanmıştır.

Bir yeteneğin değerini öğrenmek için kullanabileceğiniz üç işlev vardır ve yeteneğin değerinin türüne göre bunlardan biri kullanılır:

```
int tgetnum(char *isim);
```

işlev

tgetnum işlevi, değeri sayısal olan bir yeteneği sorgulamakta kullanılır. **isim** argümanı yeteneğin iki harflik kodu olmalıdır. Eğer belirtilen yetenek mevcutsa, **tgetnum** onun değerini negatif olmayan bir değer olarak döndürür. Eğer belirtilen yetenek uçbirim tanımında yoksa, işlev -1 değeri ile döner.

```
int tgetflag(char *isim);
```

işlev

tgetflag işlevi, değeri mantıksal olan bir yeteneği sorgulamakta kullanılır. Eğer belirtilen yetenek uçbirim tanımında mevcutsa, **tgetflag** 1 ile yoksa 0 ile döner.

```
char *tgetstr(char *isim,
              char **alan);
```

işlev

tgetstr işlevi, dizge değerli bir yeteneği sorgulamakta kullanılır. Belirtilen **isim** uçbirim tanımında mevcut bir yeteneğin ismiyse işlev yeteneğin değerine bir gösterici ile, aksi takdirde bir boş gösterici ile döner.

tgetstr işlevinin döndüreceği dizgenin saklanacağı alanı ayırmanın iki yolu vardır:

- Alanı ayırmasını **tgetstr** işlevinden talep edersiniz. *alan* argümanı ile işleve bir boş gösterici aktarırsanız, **tgetstr** yeterli alanı ayırmak için **malloc** işlevini kullanacaktır. Termcap bu alanı serbest bırakmayacağı gibi aynı alanı tekrar kullanmayacağından, işiniz bittiğinde bu alanı sizin serbest bırakmanız gerekir.

Ne kadar alan ayrılacağını düşünmeyi gerektirmediğinden bu yöntem daha sağlamsa da, sadece GNU termcap kütüphanesi tarafından desteklenmektedir.

- Gerekli alanı kendiniz ayırırsınız. *alan* argümanı olarak **char *** türünde bir göstericinin adresini vermelisiniz. **tgetstr** işlevini çağırmadan önce bu değişkeni yeterli alanla ilklendirmelisiniz. İşlev çağırıldığında dizge değerini bu alana kaydedecek ve göstericiyi bu kullanılan alanın sonrasını gösterecek şekilde arttıracaktır. Böylece, aynı göstericiyi çok sayıda **tgetstr** çağırısı yapmak için kullanabilmeniz mümkün kılınmıştır.

Tek bir dizge için ne kadar alan gerektiğini saptamanın bir yolu olmadığı gibi ne sizin tarafınızdan ne de kütüphane tarafından bellek taşmalarını önlemenin bir yolu yoktur. Ancak, bir uçbirim tanımından elde edilebilecek tüm dizge değerli yeteneklerin değerlerinin toplam uzunluğu uçbirim tanımı için ayrılan alandan daha büyük olamayacağından aynı uzunlukta bir alanı ayırmak en iyi seçim olacaktır (tabii, aynı yeteneği birden fazla sorgulamamak şartıyla). Böyle bir alanı ayırırken, belirtilecek alanın uzunluğunu **tgetent** işlevine ayrılan tamponun uzunluğu olarak **strlen** işlevi ile saptayabilirsiniz. Aşağıda böyle bir örneğe yer verilmiştir.

Gerekli alanı kendinizin ayırması gerekliliği termcap'ın Unix sürümü tarafından desteklenen tek yöntemdir.

Sorgulama işlevlerinde bir uçbirim ismi ya da bir uçbirim tanımı belirtmek zorunda olmadığınızın farkına varmış olmalısınız. Bu işlevler son yapılan **tgetent** çağırısında bulunan tanımı özdevinimli kullanırlar.

Bu örnekte, bir uçbirim tanımındaki çeşitli yetenekler sorgulanırken, koşullar kodun çalışacağı sisteme göre Unix veya GNU yöntemlerine göre ayarlanmaktadır:

```
char *tgetstr ();

char *cl_string, *cm_string;
int height;
int width;
int auto_wrap;

char PC; /* tputs için. */
char *BC; /* tgoto için. */
char *UP;

interrogate_terminal ()
{
#ifdef UNIX
/* Burada tgetent'e term_buffer'ın bizim
tarafımızdan sağlandığı varsayılmıştır. */
char *buffer
= (char *) malloc (strlen (term_buffer));
#define BUFFADDR &buffer
#else
#define BUFFADDR 0
#endif

char *temp;
```

```
/* Kullanacağımız bilgileri alalım. */
cl_string = tgetstr ("cl", BUFFADDR);
cm_string = tgetstr ("cm", BUFFADDR);
auto_wrap = tgetflag ("am");
height = tgetnum ("li");
width = tgetnum ("co");

/* termcap işlevlerinin kullanacağı bilgileri alalım. */
temp = tgetstr ("pc", BUFFADDR);
PC = temp ? *temp : 0;
BC = tgetstr ("le", BUFFADDR);
UP = tgetstr ("up", BUFFADDR);
}
```

PC değişkeni ile ilgili bilgiler için [Geciktirme](#) (sayfa: 8) bölümüne, **UP** ve **BC** için ise [Ekran Komutlarının Parametrelerle Gönderilmesi](#) (sayfa: 14) bölümüne bakınız.

2.4. Termcap Kullanımı için İklendirme

Bir uygulama yazılımı termcap'ı kullanarak komutları çıktılara başlamadan önce iki şeyi yerine getirmelidir:

- Termcap kütüphanesi çıktı işlevleri tarafından kullanılan bazı genel değişkenlerin iklendirilmesi. Bunlardan bazıları [gecikmeler](#) (sayfa: 10) için **PC** ve **ospeed** değişkenleri ve [imleç hareketleri](#) (sayfa: 15) için **UP** ve **BC** değişkenleridir.
- Çekirdeğe uçbirime yatay sekme karakterlerinin gönderilmesi ile ilgili gecikme ve değişiklik işlemlerini durdurmasının söylenmesi.

Çıktıyı kapatılması için Berkeley Unix'te **LLITOUT** isimli biti bir yapmak için **ioctl** işlevini **TIOCLSET** koduyla çağırılmalı ve **TIOCSETN** kodunu kullanarak **ANYDELAY** bitlerini temizlemelisiniz. POSIX veya System V'de ise **OPOST** isimli biti temizlemelisiniz. Daha ayrıntılı bilgi edinmek için bu sistemlerin belgelerine bakınız.

Uçbirim seçeneklerini düzgünce belirtmezseniz, bazı eski uçbirimler çalışmayacaktır. Bu, çekirdeğin çıktılamadan önce değiştirmesinin gerekli olduğu satırsonu, satırbaşı ve yatay sekme karakterlerini içeren bazı komutların bulunması sebebiyle böyledir.

Çekirdeğin uçbirim seçeneklerini değiştirdiğiniz takdirde, yazılımınızın çıkarken bunları eski duruma getirmesi önemlidir. Bunun olması için, yazılımınız **SIGQUIT** ve **SIGINT** gibi ölümcül sinyalleri yakalamalı ve asıl sonlanma öncesi bu uçbirim seçeneklerini eski durumuna getirmelidir.

Günümüz uçbirimlerinin komutları bu özel karakterleri kullanmadığından eski uçbirimlerin sorunları hakkında dikkatli olmanız gerekmeyeceğinden çekirdeğin uçbirim seçeneklerinde değişiklik yapmanız gerekmez.

2.5. Geciktirme

Geciktirme, işinin bitmesi uzun zaman alan bir uçbirim ekran komutunun ardından boş karakterlerin çıktılanması demektir. Uçbirim tanımında hangi komutların ne kadar geciktirilmesi gerektiği belirtilir. **tputs** işlevi, bir uçbirim komutunu çalıştırırken komuttan geciktirme bilgisini alır ve gerektiği kadar geciktirme uygular.

2.5.1. Geciktirme Niçin Gerekir ve Nasıl Yapılır

Çoğu uçbirim türü yüksek hızlı bir hattan gönderildiğinde, çalışması uzun süren komutlar içerir. Örneğin, ekranın temizlenmesi komutunun tamamen alınmasından itibaren 20 ms sürer. 9600 bps'lik bir hatta bağlı bir uçbirim bu süre zarfında hala ekranı temizlemekle meşgulken, çıktılamak üzere 20 ek karakter alabilirdi. Her uçbirimin,

henüz işleme alınmamış karakterleri hatırd tutmak için kullandığı belli bir miktarda tamponlama kapasitesi vardır. Fakat çok yavaş komutlarda tek bir satır bile bu tamponu doldurmak için yeterli olur. Bunların ardından gelen komutların saklanacağı bir yer kalmadığından bu komutlar kaybolup gider.

Bu sorunla karşılaşmamak için, ekran komutunun gerektirdiği çalışma süresince hattan işe yaramaz karakterler (genellikle boş karakter) göndeririz. Eğer uçbirim çoğu uçbirimin yaptığı gibi bu boş karakterleri tamponu doldurmasın diye yoksayıyorsa, istenen etki sağlanmış olur. Eğer yeterince geciktirme sağlanabilmişse hiçbir çıktı kaybı olmaz. Doğru geciktirme, işlemin yavaşlamasına sebebiyet vermeksizin çıktı kaybını engelleyen geciktirmedir.

Geciktirme amacıyla hattan gönderilecek boş karakterlerin sayısı hattın hızına göre değişir. 9600 bps'lik bir hattan her ms'de bir karakter gönderilebilir, yani ekranın temizlenmesi için gerekli 20 ms için 20 boş karakter göndermek yeterli olur. 1200 bps'lik bir hat için bu sadece üç karakterlik bir geciktirme demektir.

2.5.2. Geciktirme Yetersiz Kalırsa

Yetersiz geciktirmenin iyi bilinen bazı göstergeleri vardır.

- Emacs ekranın altında **I-search: ^Q-** çıktılar.

Bu uçbirim tamponunun ekran komutları ile tamaen dolduğu anlamına gelir, böylece bilgisayara artık birşey göndermemesini söylemeyi denemiş olur.

- Ekran fasıllı olarak bozunmaya uğrar ya da siz eylemi yinelerken bozunmanın ayrıntıları değişir. (Bozunmaya uğramış bir ekranın başlıca sebebi bir komutun basitçe yanlış uygulanması ya da kullanıcının seçimlerinin uçbirim tanımındaki kabullerle uyumsuz olmasıdır ki, bundan genelde geridönüş yoktur – kilitleme.)

Bu, tamponun dolmadığı halde bazı komutların kaybedilmesi demektir. Hangi komutun kaybedileceği çok sayıda olasılığa bağlı olarak değişiklik gösterir.

- Ekran yüksek hızlarda bozunmaya uğrarken düşük hızlarda bu olmaz. Geciktirme sorunları hemen hemen daima hız azaltılınca kaybolur, 1200 baud için bile.

Bu, daha yüksek bir hızda komutların icra süresinden daha hızlı alınması demektir.

Esrarengiz bir uçbirim üzerinde her esrarengiz komut geciktirme sorunlarına yol açabilir. Pratikte bu sorunlarla karşılaşılan komutlar çoğunlukla, **cl** ve **cd** *ekran temizleme komutları* (sayfa: 28), **sf** ve **sr** *ekran kaydırma komutları* (sayfa: 25) ile **al** ve **dl** *satır ekleme ve satır silme komutlarıdır* (sayfa: 28).

Bazan uçbirim tanımı **sf** tanımında başarısız olur, bu durumla karşılaşmamak için bazı uygulamalar bunun yerine **do** kullanır, yani sorun karşınıza **do** ile ilgili olarak çıkabilir. Böyle bir durumda, **do** için olduğu gibi önce **sf**'yi tanımlar ardından da **sf**'ye biraz geciktirme eklersiniz.

En iyi strateji baştan 200 ms'lik bir geciktirme eklemektir. Bu yeterince fazladır; aslında, gözle görünür bir yavaşlamaya sebep olması gerekir. (Eğer böyle bir yavaşlık görmüyorsanız, değişiklik etkin olmamıştır; bkz. *Etkin Veritabanı Değişirse* (sayfa: 19).) Eğer bu sorunu gideriyorsa, geciktirmeyi uygulayacak doğru yeri bulmuşsunuz demektir; artık sorun ortaya çıkana kadar geciktirmeyi azaltabilirsiniz, sorun ortaya çıkınca da tekrar arttırırsınız. Eğer sorun devam ediyorsa ya sorun başka bir yetenekle ilgilidir ya da yetenek için geciktirme anlamlı değildir.

Çoğu uçbirim imleç konumuna bağlı ekran kaydırma ve satır ekleme/silme için doğru geciktirmelere sahip olduğundan o kadar tasalanmayın. Eğer ekranın büyükçe bir bölümünü kaydırırken bu sorunla karşılaştığınız halde küçük bir bölümünü kaydırırken sorun çıkmıyorsa, bu, konuma bağlı gecikme yerine sabit gecikme kullanmanız gerekli olduğu anlamına gelebilir.

2.5.3. Uçbirim Tanımında Geciktirmenin Belirtilmesi

Uçbirim tanımındaki, her ekran komutunun başında kendi için gerekli geciktirme miktarını belirten bir dizi rakam bulunur. Bu rakamlar geciktirme süresinin ms cinsinden ifadesidir. Bu rakamları isteğe bağlı olarak bir nokta ve bir rakam izleyebilir; bu, gerektiğinde onda bir milisaniyeleri belirtebilmek için kullanılır.

Bazan komutun ihtiyaç duyduğu geciktirme imlecin konumuna bağlıdır. Örneğin, satır girme komutunun gerektirdiği süre bir satırın aşağı taşınıp boş bir satır elde edilmesi için kullanıldığından ihtiyaç duyulan süre satır sayısına bağlıdır. Geciktirme süresinden sonra bir yıldız işareti (*) koyarak geciktirme süresinin komuttan etkilenen satır sayısının katları oranında arttırılacağını belirtmiş oluruz.

```
:al=1.3*\E[L:
```

Bu, bir uçbirim için satır girme komutunun tanımı olsun. Buradaki geciktirme süresi her satır için 1.3 milisaniyedir. Komutun kendisi ise **<ESC> [L**'dir.

Bu şekilde belirtilmiş geciktirme süresi **tputs**'a çıktıya kaç ek karakter göndereceğini belirtir. Bkz. [tputs ile Geciktirme](#) (sayfa: 10).

Tüm komutlar için geciktirmeyi etkileyen iki özel yetenek vardır. Bunların isimleri **pc** ve **pb**'dir. **pc** değişkeni geciktirme için kullanılacak karakteri, **pb** ise geciktirmenin gerekmediği hız değerini içerir. Bu değişkenler için öntanımlı değerler sırasıyla boş karakter ve 0'dır. Bu değerler çoğu uçbirimde böyledir. Bkz. [Geciktirme Yetenekleri](#) (sayfa: 40).

2.5.4. tputs ile Geciktirme

Önceki bölümlerde açıklanan geciktirme işlemini (sayfa: 9) gerçekleştirmek için bir termcap işlevi olan **tputs** kullanılır. **tputs** işlevi komutun tanımından geciktirme belirtimini ayırdıktan sonra komutu çıktıya gönderir; bu dizgenin hemen ardından geciktirme belirtimine uygun sayıda boş karakteri gönderir.

```
int tputs(char *dizge, int satir_sayisi, int (*çikti_işlevi) ());
```

Burada *dizge* çıkılacak dizge (geciktirme belirtimini de içerir); *satir_sayisi* işlemde etkilenen satır sayısı (eğer komutun tanımı * ile biten bir geciktirme belirtimi içeriyorsa, bu değer geciktirme süresi ile çarpılır). Son argüman olan *çikti_işlevi* ise argüman olarak tek karakter alacak (**fputc** gibi) ve geciktirme için gönderilecek karakterleri çıkılacak işlevin ismi olmalıdır.

tputs ile ilgili işlemlerin denetimi **ospeed** ve **PC** isimli iki değişkenle yapılır. **ospeed** değeri uçbirimin çıkılma hızı kabul edilir, **ioct1** sistem çağrısı onunla kodlandığında hız bilgisi ile döner. Bu değer geciktirme amacıyla yollanacak karakterlerin sayısını hesaplamakta kullanılır. **PC** değişkeni ise geciktirme amacıyla kullanılacak karakteri içerir.

tputs işlevini kullanmadan önce bu değişkenleri uygun değerlerle ilklendirmiş olmalısınız. **PC** değişkeninin değeri uçbirim tanımındaki **pc** yeteneğinden elde edilmelidir (bkz. [Geciktirme Yetenekleri](#) (sayfa: 40)). Tanım bir **pc** yeteneği içermiyorsa bu değişkeni 0 ile ilklendirin.

satir_sayisi argümanı biraz daha fazla ilgi gerektirir. Normalde, komut tarafından yeri değiştirilen ve boşaltılan satır sayısıdır. İmleç hareket komutlarında ya da tek satırlık düzenleme komutlarında 1 değeri kullanılır. **al** (bir satır gir) ve **cd** (imleçten ekranın sonuna kadar sil) gibi birden fazla satırı etkileyen komutlarda ise bu değer ekran yüksekliği eksi o anki düşey konum olmalıdır (başlangıç 0'dır). **AL** (çok sayıda satır ekle) gibi çok sayıda satır ekleme ve ekran kaydırma komutlarında da değer bu şekile saptanır; girilen satırların sayısı olarak "değil".

Bir komut tarafından etkilenen satırların sayısını azaltmak için pencere kaydırma özelliği kullanılıyorsa, *satir_sayisi* değeri bunu hesaba katmalıdır. Bu, geciktirme süresinin uçbirimin ne kadar süreyle meşgul

olacağına bağlı olmasından dolayıdır ve pencere kaydırma özelliği bu süreyi kısaltır. Bkz. *Kaydırma* (sayfa: 25).

ic ve **dc** komutları da (karakter ekleme ve silme komutları) imleç konumundan satır sonuna kadar sütun sayısı olarak bu komuttan etkilenen karakter sayısına bağlı olarak geciktirme gerektirmeleriyle biraz sorunludurlar. Bu bağımlılığın belirtilmesini mümkün kılacak bir yol olsaydı iyi olurdu, böylece bu komutlar için *satır_sayısı* değerinin aslında etkilenen sütun sayısı olduğunu söylemek zorunda kalmazdık. Ancak, termcap tanımına göre *satır_sayısı* daima etkilenen satırların sayısıdır, bu durumda bile bu böyledir ve bu durum için bu değer 1'dir. Bu kuralı artık değiştirmek kolay değildir, çünkü artık yazılmış çok fazla uygulama ve uçbirim tanımı var.

ic ve **dc** dizgeleri için *satır_sayısı* daima 1 olduğundan, bunlar için * kullanmak anlamlı değildir ama yine de bazıları bunu yapıyor. Bunlar * imi silinerek düzeltilmelidir. Eğer bir gün böyle girdiler kullanılmamaya başlarsa, herhangi bir yazılımı etkilemeksizin bu işlemler için *satır_sayısı* argümanı yerine daha elverişli bir uzlaşım kullanmak mümkün olabilir.

2.6. Parametreler

Bazı uçbirim denetim dizgeleri sayısal *parametreler* gerektirir. Örneğin, imlecin yerini değiştirirken, onun taşınacağı yatay ve düşey konumları belirtmeniz gerekir. imlecin nasıl taşınacağını belirleyen uçbirimin **cm** yeteneğinin değeri basitçe bir dizge değeri olamaz; imleç konumlarının nasıl ilerletileceği komutun onu nereye koyacağı belirtilmelidir.

Termcap belirtimi, hangi dizge değerli yeteneklerin kaç tane parametre gerektirdiği ve bu parametrelerin anlamları gibi uzlaşım içerir. Örneğin, belirtim, **cm** dizgesi için üst sol köşe 0,0 olmak üzere düşey ve yatay konumları belirten iki parametre alır, der. Bu uzlaşım her komut için belgede açıklanmıştır.

Termcap ayrıca, yetenek belirtimlerinde kullanılmak üzere çıktı parametrelerinin nerede nasıl kodlanacağını belirleyen bir dil de tanımlamıştır. Bu dil % karakteri ile başlayan karakter dizileri kullanır. (Fikir **printf**'deki ile aynı olduğu halde, ayrıntılar bakımından biraz farklıdır.) Dilin parametre kodlaması ile ilgili kısmı bu bölümde anlatılacaktır.

Ekranı çıkıtılama yaparken belirtme göre parametreleri kodlamak için bir yazılım **tparam** ya da **tgoto** işlevini çağırır. Bu işlevler çıkıtılacak komutu içeren bir dizge üretirler (keza, bir geciktirme belirtimi de **tputs** ile işlenir; bkz. *Geciktirme* (sayfa: 8)).

2.6.1. Kodlamanın Açıklaması

Parametre gerektiren bir uçbirim komut dizgesi, bu parametrelerin nasıl kodlanacağını bir belirtimi olarak % karakteri ile başlayan özel karakter dizileri içerir. Bu diziler **tparam** ve **tgoto** işlevlerinin eylemlerini denetim altına almak için kullanılır.

tparam ve **tgoto** işlevlerine aktarılacak parametre değerlerinin bir vektör biçiminde olacağı varsayılır. Bu vektör içindeki bir gösterici işlenecek sonraki parametreyi belirler. %'li dizilerin bazıları tek bir parametreyi kodlar ve göstericiyi sonraki parametreye iletir. Bazıları da çıktı üretmeksizin ya göstericiyi ya da parametre değerini değiştirir.

Örneğin, bir standart ANSI uçbiriminin **cm** dizgesi **\E [%i%d; %dH** biçiminde yazılır. (**\E**, **<ESC>**'e karşılıktır.) **cm** teamülen daima iki parametre gerektirir: düşey ve yatay konum; yani bu dizge iki parametrenin kodlanmasını gerektirir. Burada **%i** kendinden sonra gelen iki değeri 1 arttırır. **%d**'ler ise değerleri birer onluk sayı olarak kodlar. Eğer imleç konum değerleri 20,58 olarsa bu dizge, sonucu **\E [21; 59H** olacak şekilde kodlanmalıdır.

Öncelikle, bunlar çıktı üreten % dizileridir. %% dışında herbiri bir parametreyi kodlayıp göstericiyi sonraki parametreye iletir.

%%

Tek bir % çıktılar. Sadece % işaretini çıkıtlamak içindir. %% bir parametre olarak değerlendirilmez.

%d

printf işlevindeki gibi parametresini onluk sayı olarak çıktılar.

%2

printf işlevindeki **%02d** gibidir: parametresini onluk sayı olarak çıktılar ve daima an az iki rakam kullanır.

%3

printf işlevindeki **%03d** gibidir: parametresini onluk sayı olarak çıktılar ve daima an az üç rakam kullanır.



Bilgi

%4 ve daha büyük diziler tanımlanmamıştır.

%.

Parametre değerini tek bir karakterin ASCII kodu olarak yorumlar ve karakteri çıktılar. **printf** işlevindeki **%c** gibidir.

%+*karakter*

Parametre değerini *karakter* karakterine ekleyerek sonucu çıktılar. Örneğin, '+ ' ile 0 değeri boşluk, 1 değeri ! olarak çıktılanır.

Aşağıdaki % dizileri bir parametreyi çıkıtlamazlar parametrelerde (değer ya da sıralarında) değişikliğe sebep olurlar, yani sadece yan etkilerinden yararlanılır. Ayrıca gösterici konumunu açıkça belirtilmedikçe ilerletmezler. Standart Unix termcap'i sadece %i, %r ve %> dizilerini tanımlamıştır. Diğerleri GNU oluşumudur.

%i

Sonraki iki parametrenin değerlerini 1 artırır. Bu imleç başlangıcını 0 değilde 1 kabul eden uçbirimlerde kullanılır. Örneğin, %i%d, %d iki parametreden 0 olanını 1, 1 olanını 2 olarak çıkıtlayacaktır.

%r

Sonraki iki parametrenin yerlerini değiştirir. İmlecın yeri belirtilirken yatay konumun önce belirtilmesi gereken uçbirimlerde yararlıdır.

%s

Sonraki parametrenin atlanmasını sağlar. Bir çıktı üretmez.

%b

Bir parametreyi yedekler. Bu parametre son parametre ise tekrar çıkıtlanacak son parametre haline gelecek ve sonraki çıktı komutu tarafından kullanılacaktır. %b'i birden fazla kullanarak çok sayıda parametreyi hatta bir parametreyi defalarca yedekleyebilirsiniz.

%>c1c2

Sonraki parametreyi koşullu artırır. Burada c1 ve c2 ASCII kodları sayısal olarak verilmiş iki karakterdir. Eğer sonraki parametre c1'in ASCII kodundan büyükse, parametreye c2'nin ASCII kodu eklenir.

%a *işleç tür konum*

Kendinden sonraki parametreye bir aritmetik işlem uygular. Burada *işleç* aritmetik işlemin işleci, *tür* ve *konum* ise ikisi birlikte, işleme sokulacak "diğer terimi" belirtir. Herhangi bir çıktı üretilmez.

İfade içindeki boşluklar terimleri ayırmak içindir; bu boşluklar veritabanında görünmez, sadece 5 karakter uzunlukta bir dizi bulunur.

işleç karakteri uygulanacak aritmetik işlem çeşidini belirten bir işleçtir. Şu karakterlerden biri olmalıdır:

=
Sonraki parametreye "diğer terimi" atar, eski değer yoksayılır.

+
Sonraki parametreye "diğer terimi" ekler.

-
Sonraki parametreden "diğer terimi" çıkarır.

*
Sonraki parametre ile "diğer terimi" çarpar.

/
Sonraki parametreyi "diğer terime" böler.

"Diğer terim" bir sabit olabileceği gibi, başka bir parametre de olabilir; *tür* karakteri bunu belirtmek içindir. Şunlardan biri olabilir:

p
Başka bir parametre kullanılır. *konum* ile hangi parametrenin kullanılacağı belirtilir. *konum* karakterinin ASCII kodundan 64 çıkarıldığında kalan, "diğer terim" olarak kullanılacak parametrenin bu parametreye göre (görelî) konumunu vermelidir. *konum* olarak **A** karakteri verilmişse "diğer terim" olarak sonraki parametreden sonraki parametre kullanılır. Karakter **?** ise sonraki parametreden önceki parametre kullanılır.

c
Değerin bir sabit olduğunu belirtir. *konum* karakteri ile sabitin değeri belirtilir. 0200 biti boştur, dolayısıyla 0200 sıfır için kullanılabilir.

Aşağıdaki % dizileri esrarengiz uçbirimlerin tuhaf tasarımlarına imkan vermek için kullanılan özel amaçlı dizilerdir. Genellikle sonraki bir ya da iki parametre üzerinde işlem yaparlar, başka parametre kullanmazlar ve bir çıktı üretmezler. **%m** bir GNU oluşumdur, diğerleri standart Unix termcap'ta tanımlıdır.

%n
Sonraki parametreye 0140 ile "ayrıcalıklı veya" uygular, ondan sonraki parametreye de aynısını yapar.

%m
Sonraki parametrenin bitlerinin eşleniğini alır, bunu sonraki parametreye de uygular.

%B
Sonraki parametreye BCD kodlaması uygular. Bu C deyimini yeni değerin nasıl hesaplandığını göstermektedir:

```
parm = (parm / 10) * 16 + parm % 10;
```

%D
Delta Data uçbirimlerin gerektirdiği dönüşümü sonraki parametreye uygular. Bu C deyimini yeni değerin nasıl hesaplandığını göstermektedir:

```
parm == 2 * (parm % 16);
```

2.6.2. Ekran Komutlarının Parametrelerle Gönderilmesi

Termcap kütüphanesinin **tparam** ve **tgoto** işlevleri uçbirim dizge parametreleri için **printf** işlevinin yaptığına benzer şekilde kullanılabilir. Yeni **tparam** işlevi bir GNU oluşumdur, daha genel amaçlıdır ancak, Unix termcap'ta yoktur. İmleç hareketleri için bir parametre kodlama işlevi olarak **tgoto** tercih edilen bir işlevdir.

tparam işlevi herhangi bir sayıdaki parametreye sahip ekran komutlarını kodlayabilir ve bir tampon alanı belirtmenizi mümkün kılar. **cm** yeteneği dışında tüm yeteneklerin parametreleri için tercih edilen bir işlevdir.

```
char *tparam(char *biçim_dizgesi, işlev
               char *tampon,
               int boyut,
               int parm1,
               ...);
```

Burada, *biçim_dizgesi*'nin bir uçbirim yeteneğinin değeri olacağı varsayılır; *tampon* ve *boyut* ise çıktı tamponu içindir. *parm1*, ... ise biçim dizgesinde kodlanacak tamsayı parametrelerdir. **tparam** işlevi biçim dizgesi içindeki % dizilerini parametrelerle kodlayarak sonucu tampona kopyalar.

Çıktı tamponunu adresiyle *tampon* argümanında, bayt cinsinden boyuyla *boyut* argümanında siz tanımlayabilirsiniz. Eğer ayırdığınız alan yetersiz kalırsa, **tparam** işlevi **malloc**'u kullanarak tamponu büyütür. Her iki durumda da **tparam** tamponun adresi ile döner. Eğer dönüş değeri *tampon* ile aynıysa sizin ayırdığınız alan kullanılmıştır, değilse, işlev yeni bir tampon ayırmıştır ve onu da çıktılama tamamlanınca siz serbest bırakmalısınız. Eğer işleve *boyut* ve *tampon* olarak 0 değeri aktarmışsanız, tampon için gerekli alanı **tparam** kendisi ayırır.

Parametre gerektirmekten başka geciktirme belirtilmesi zorunlu olan yeteneklerde **tparam** tarafından üretilen çıktı için **tputs** kullanmalısınız. Bkz. [Geciktirme](#) (sayfa: 8). Burada buna bir örnek verilmiştir:

```
{
char *buf;
char buffer[40];

buf = tparam (command, buffer, 40, parm);
tputs (buf, 1, fputchar);
if (buf != buffer)
free (buf);
}
```

Eğer sıfır değerli bir parametre % . tarzı kodlama ile kodlanırsa, **tputs** buna memnun olmayacak ve bu bir sürü soruna yol açacaktır. Fakat, şansa bakın ki, % . tarzı kodlama birkaç modası geçmiş uçbirim tarafından sadece **cm** yeteneğinde kullanılmaktadır. Bu sorunla karşılaşmamak için **cm** yeteneğini kodlarken **tparam** işlevini değil **tgoto** işlevini kullanın.

İmleç hareketlerinde bazı özel durumlar **tgoto** işlevinin kullanılmasını gerektirmektedir. **tgoto** kullanmanızı gerektiren sebepler şunlardır:

- Unix uyumluluğu için. Çünkü Unix termcap, **tparam** işlevine sahip değildir.
- **cm** yeteneği için. **tgoto** işlevi, parametre kodlamasında bazı eski model uçbirimlerdeki boş karakter, sekme ve satırsonu karakteri ile ilgili sorunları önlemek amacıyla bu yetenek için % . kodlamasını kullanır.

```
char *tgoto(char *cm_dizgesi,                                     işlev
             int  yatay_konum,
             int  düşey_konum);
```

İşlevin üç argümanı vardır: uçbirim tanımının **cm** dizgesi ile imlecin yatay ve düşey konumları. **tgoto** parametrelili dizgeyi bir iç tamponda kendisi hazırlar ve bu tamponun adresi ile döner. İşlevin daha sonraki kullanımlarında **tgoto** aynı tamponu yeniden kullanacaktır.

%. kodlaması ile kodlanan parametreler boş karakterler, sekmeler ve satırsonu karakterleri üretebilir. Bunlar sorunlara yol açabilir: Boş karakterler **tputs**'un dizgenin sonuna gelindiğini varsaymasına sebep olabilir. Sekmeler, çekirdeğin veya başka bir uygulamanın bu karakteri boşluklarla genişletmesine sebep olabilir. Satırsonu karakterlerinde ise çekirdek bir satırbaşı karakteri ekleyebilir ya da satırsonunu aşmak için normal olarak satırın sonunu boşluklar ekleyebilir. Bu tür sorunlarla karşılaşmamak için **tgoto** bu karakterlerden kaçınmada dikkatlidir. Pek iyi de, bunu nasıl yapar: Hedef imleç konumu (sıfır, dokuz ya da on) böyle bir soruna yol açacak gibiyse, **tgoto** konumu bir artırır ve böylece imlecin bir dizge eklenerek bir konum geri ya da yukarı hareket etmesine engel olur (sorunu telafi eder).

Geri ya da yukarı taşıma için kullanılacak telafi dizgeleri **BC** ve **UP** isimli genel değişkenlerde bulunur. Bunlar büyük harfli isimleriyle aslında birer **external C** değişkenleridir ve **char *** türünde bildirilirler. Uygun değerleri siz belirtirsiniz. Bu değerler normalde, uçbirim tanımından **tgetstr** işlevi kullanılarak **le** ve **up** uçbirim yeteneklerinden elde edilir. Ayrıca, bu iki değişken sıfır da olabilir, bu durumda boş karakterler, sekmeler ve satırsonu karakterleri ile ilgili sorunların telafisine çalışmaz.

BC ve **UP** değişkenlerinin değerleri sıfır olduğunda **tgoto** işlevi sadece **cm** dışında, diğer tüm komutlar için güvenlidir.



Bilgi

tgoto işlevinde yatay ve düşey konum argümanları ters sıradadır. Hatta **cm** dizgesinde bile düşey konum parametresi yatay konum parametresinden öncedir. Eğer **tgoto** işlevini **AL** gibi tek parametre alan bir komut ile kullanıyorsanız, parametreyi *düşey_konum* argümanı ile işleve aktarmalısınız.

3. Veritabanının Biçimi

Uçbirim tanımının termcap veritabanı **/etc/termcap** dosyasında saklanır. Bu dosya uçbirim tanımları, boş satırlar ve açıklamalardan oluşur.

Bir uçbirim tanımı uçbirim türü için bir ya da daha fazla isim ile başlar. Tanım bilgisi ardarda sıralanmış **yetenek isimleri** ile bunların değerlerinden oluşur. Uçbirimi, anlamları standart olan **yetenek isimleri** (sayfa: 19) ve bunların değerleri tanımlar.

3.1. Uçbirim Tanımının Biçimi

Termcap veritabanında, açıklamalar (**#** ile başlayan satırlardır ve yoksayıllırlar) dışında, boş olmayan her satır bir uçbirim tanımdır. Bir uçbirim tanımı normalde tek bir satırdan oluşur, ancak satırsonu karakterinden önce bir **** ekleyerek tanım çok sayıda satıra bölünebilir. Eklenen bu iki karaktere bir **\ satırsonu** dizgesi olarak bir tanım içinde rastlanırsa bu dizgeler yoksayıllırlar (yani böyle satırlar tek bir satır gibi ele alınır).

Tercih edilen yöntem tanımı yeteneklerine göre bölümlere ayırmaktır: Her sütunun hemen önüne dört karakter, **: \ satırsonu sekme** yerleştirilir. Böylece her alt satırın aynı girinti miktarı ile başlaması sağlanmış olur. **\ satırsonu** dizgeleri yoksayıldığından geriye **: tab :** kalır. Bir yetenek ilk iki nokta üstüste karakterinde

bitirken diğer yetenek ikincisinden başlar ve bu böyle gider. Eğer sadece `\ satırsonu` ile ayırırsanız ardına bir girinti ekleyemeyebilirsiniz.

Gerçek bir uçbirim tanımı örneği:

```
dw|vt52|DEC vt52:\
:cr=^M:do=^J:nl=^J:bl=^G:\
:le=^H:bs:cd=\EJ:ce=\EK:cl=\EH\EJ:\
:cm=\EY%+ %+ :co#80:li#24:\
:nd=\EC:ta=^I:pt:sr=\EI:up=\EA:\
:ku=\EA:kd=\EB:kr=\EC:kl=\ED:kb=^H:
```

Her uçbirim tanımı uçbirimin türünü ifade eden isimlerden oluşan bir liste ile başlar. İsimler birbirlerinden `|` karakteri ile ayrılır ve liste bir iki nokta üstüste karakteri ile sonlandırılır. İlk isim uçbirimin kısa ismidir ve iki karakter uzunlukta olmalıdır, bu isim sadece çok eski Unix sistemlerinde kullanılırdı; günümüz sistemlerinde kullanılmamaktadır. Listedeki son isim ise "DEC vt52" veya "Ann Arbor Ambassador with 48 lines" gibi uçbirimin tam ismi olmalıdır. Diğer isimler kullanıcının bu uçbirim türünü kullanmak istediğinde belirteceği isimler olmalıdır, **vt52** veya **aaa-48** gibi. Uçbirim isimlerinin nasıl seçildiği [Uçbirim Türü İsimlendirme Uzlaşmaları](#) (sayfa: 17) bölümünde açıklanmıştır.

Uçbirim isimlerinden sonra uçbirim yetenekleri listesi gelir ve bunlar da iki nokta üstüste karakterleri ile ayrılırlar. Her yeteneğin iki harften oluşan bir ismi vardır; örneğin, ekranın satır sayısı için **li**, imleç konumlandırma dizgesi için **cm** gibi.

3.2. Yeteneklerin Yazılması

Yetenekler aldıkları değere göre üç çeşittir: dizgeler, sayılar ve seçimlikler. Her birinin tanım içindeki sözdizimleri kendilerine özeldir ve her yeteneğin uzlaşım ile belirlenmiş belli değer çeşidi vardır. Örneğin, **li** yeteneği daima sayısal bir değere sahipken, **cm**'nin değeri daima bir dizgedir.

Bir seçimlik yetenek bir mantıksal değer içerir: eğer bu yetenek mevcutsa değeri doğru, değilse yanlış'tır. Yetenek mevcutsa sadece ismi yazılır, yoksa yazılmaz (yani bu yetenekler aslında bir değer almaz, sadece isimlerinin uçbirim tanımında var olup olmadıklarına bakılır).

Sayısal değerli bir yetenek daima negatif olmayan bir değere sahiptir. Yetenek isminden sonra bir **#** konup sayısal değer yazılır. Örneğin, **li** yeteneği `...:li#48:...` şeklinde yazılmışsa ekranda 48 satır gösterme yeteneğinde bir uçbirim sözkonusudur.

Dizge değerli bir yetenek daima bir karakter dizisi içerir. Genellikle bunlarda ekranda bazı işlemler gerçekleştiren karakterler kullanılır. Yetenek isminden sonra bir **=** konup karakterler yazılır. Örneğin, **cm** yeteneği `...:cm=\E[%i%d;%dH:...` şeklinde bir komut ve bu komutun biçim dizgesi olarak yazılır ve aldığı parametreler imlecin ekranda konumlanacağı yeri belirler.

Bir dizge değerindeki özel karakterler C dilindeki gibi `\` ile öncelenerek yazılırlar. `\E` dizgesi `<ESC>`'e karşılıktır. `^` karakteri bir tuşun Ctrl tuşu ile birlikte basıldığı durumu ifade etmek için kullanılır. Örneğin, `^a` dizgesi Ctrl-a demektir ve `\001` ile aynıdır. `\` ve `^` karakterleri sıradan birer karakter olarak yazılmak istenirse `\\` ve `^^` şeklinde yazılırlar.

Bir dizge içinde bir iki nokta üstüste karakteri kullanmak isterseniz onu `\072` biçiminde yazmalısınız. Diyeceksiniz ki, "neden onu da `\:` biçiminde yazmayalım?"; sebebi, bir yeteneği uçbirim tanımında ararken sorgulama işlevlerinin tersbölü karakterlerini görmezden gelmeleridir. `:ce=ab\:` biçiminde yazılmış bir **ce** yeteneğinin **ab:cd** değerine sahip olacağını düşünürseniz yanılırsınız, **cd** bir seçimlik yetenek olarak ele alınacaktır.

Dizge değerlerin çoğu [geciktirme miktarını](#) (sayfa: 8) ifade eden bir takım rakamlarla başlar ve dizge içinde `%` karakteri ile başlayan ve [parametrelerin](#) (sayfa: 11) nasıl kodlanacağını belirten alt dizgeler içerirler. Bunlar

uçbirime doğrudan çıktılanmazlar, yeteneğin değerinin bir parçası olarak **tputs**, **tparam** veya **tgoto** işlevleri tarafından özel olarak ele alınırlar. Daha özel olarak, \ ve ^ dizgedeki karakterleri belirtmek için sözdiziminin parçaları olarak yorumlanırlar.

VT52 örneğine tekrar bakalım:

```
dw|vt52|DEC vt52:\
:cr=^M:do=^J:nl=^J:bl=^G:\
:le=^H:bs:cd=\EJ:ce=\EK:cl=\EH\EJ:\
:cm=\EY%+ %+ :co#80:li#24:\
:nd=\EC:ta=^I:pt:sr=\EI:up=\EA:\
:ku=\EA:kd=\EB:kr=\EC:kl=\ED:kb=^H:
```

Buradaki yeteneklerden **co** ve **li** sayısal değerli, **bs** ve **pt** ise seçimlik yeteneklerdir. Diğerleri dizge değerli yeteneklerdir. Dizgelerin çoğu <ESC>'i ifade eden **\E** ile başlarken, kalanı denetim karakterlerini ifade eden ^ karakterini içeriyor. Tek tek yeteneklerin anlamları bu tanımda yer almaz, onlar başka yerdedir (bkz. [Uçbirim Yeteneklerinin Tanımları](#) (sayfa: 19)).

3.3. Uçbirim Türü İsimlendirme Uzlaşmaları

Uçbirim türünün seçiminde kullanılan bazı uzlaşmalar vardır. Bunlardan biri, tüm harflerin küçük harf olması gerekliliğidir. En alışılmış, en temel işlem kipindeki bir uçbirim için uçbirim ismi bir tire içermemelidir.

Eğer aynı uçbirimin farklı uçbirim tanımları gerektiren başka işlem kiplerine sahip sürümleri varsa, bunların isimlerinin başına, bir tire ile ayırarak temel işlem kipindeki uçbirimin ismi yazılır. Böyle ikincil tanımların kullanımı için iki sebep vardır:

- Uçbirim davranış değişikliği sağlayan bir seçeneğe sahiptir. Sistem bu seçeneğin nasıl etkin kılınacağını bilmediğinden kullanıcı bunu uygun uçbirim türünü seçerek yapabilmelidir.

Örneğin, VT-100 imlecin sağ kenardayken bir alt satıra geçip geçmeyeceğini denetleyen bir ayar seçeneğine sahiptir. Eğer imlecin sağ kenara geldiğinde alt satıra geçmesini istiyorsanız **vt100-am**, aksi takdirde **vt100-nam** uçbirim türünü seçmelisiniz. Düz **vt100** sistem yöneticisinin tercihine bağlı olarak ya **vt100-am** ya da **vt100-nam** ile eşanlamlı uçbirim türü olarak tanımlanır.

Standard **-am** soneki, imleç sağ kenara geldiğinde alt satırın başına gidecek anlamındadır.

- Kullanıcıya kendi uçbirimini seçme imkanı vermek için. Uçbirim, sistemin denetleyebildiği bir seçeneğe sahip olduğunda bu yapılabilir.

Örneğin, Ann Arbor Ambassador 20 ila 60 satırlık bir ekran için yapılandırılabilir. Satırlar azaldığında daha büyük karakterler kullanılırken, arttırıldığında düzenlemekte olduğunuzdan daha fazla satır görürsünüz. Sonuç olarak kullanıcı farklı tercihler yapabilir. Bununla birlikte, termcap çeşitli ekran boyutlarına uygun uçbirim türleri sağlar. Eğer **aaa-30** uçbirim türünü seçerseniz, uçbirim 30 satırlık, **aaa-48** seçerseniz 48 satırlık yapılandırılacaktır.

Standart sonekler ve uzlaşmış anlamları:

-w

"wide" (geniş) kısaltması. Bu kip uçbirimin daha çok sütun içermesini sağlar. Bu normalde bir kullanıcı seçeneğidir.

-am

"Automatic margins" (imleç sağ kenerdan alt satırın başına gidecek) kısaltması. Uçbirimin bu yeteneğe sahip olması isteniyorsa bu soneki içeren tür seçilmelidir. Bu seçenek normalde (soneksiz isme sahip uçbirim türünde) kapalıdır.

-nam

"No automatic margins" kısaltması. **-am** kısaltmasının zıddıdır. Normal uçbirimde **am** yeteneği etkinse, bu sonek, bu yeteneğin kapalı olduğu uçbirim türünü seçmek için kullanılır.

-na

"No arrows" (ok tuşları yok) kısaltması. Bu uçbirim tanımı ok tuşlarını yerel kipte etkisiz olarak ilklendirir. Bu bir kullanıcı seçeneğidir.

-rv

"Reverse video" kısaltması. Ekran koyu zemin üzerinde açık renk metin göstermek yerine açık renk zemin üzerinde koyu renk metin gösterecek şekilde ya da tam tersi (normal uçbirim tanımında yeteneğin etkin olup olmamasına bağlı olarak) elde edilecek şekilde yapılandırılır. Bu seçenek iki dizgenin bir diğerine göre zıt renklerde gösterilebilme yeteneğiyle ilgili olarak kullanılmaz.

Bu bir kullanıcı seçeneğidir. Hangi tür uçbirimi seçerseniz seçin termcap buna itaat edecektir.

-s

"Status" kısaltması. Normal çıktının dokunmadığı bir durum satırının etkinleştirilmesi içindir. (Bkz. *Durum Satırı* (sayfa: 41)).

Bazı uçbirimler sadece durum satırı olarak kullanılabilen özel bir satıra sahiptir. Bu tür uçbirimlerin bir **-s** sonekine ihtiyacı yoktur; durum satırı komutu öntanımlı tanımlanmıştır. Diğer uçbirimlerde ise bir durum satırını etkinleştirmek demek, ekrandan bir satırı normal kullanımdan kaldırmak, etkin ekran yüksekliğini bir satır eksiltmek demektir. Bu tür uçbirimlerde kullanıcı durum satırı kullanmak istediğinde uçbirimi seçerken **-s** soneklisini tercih edecektir.

-satur_sayısı

Ekranın *satur_sayısı* satırlık olacağını belirtir. Ambassador gibi bazı uçbirim türleri bu seçeneğe sahiptir. Normalde bu bir kullanıcı seçeneğidir; uçbirim türünü seçerek termcap'in kaç satır kullanacağını belirleyebilirsiniz.

-sayfa_sayısıp

Uçbirimin kaç *sayfa_sayısı* sayfalık ekran belleği olacağını belirler. Uçbirimler için bu bir donanım seçeneğidir.

-unk

Uçbirim tanımının doğrudan kullanıma uygun olmadığını belirtir. Bu tür tanımlar **tc** yeteneği sayesinde başka uçbirimlerin tanımlarına eklenirler. **-unk** soneki yerine istediğiniz yetenekleri sağlayacak uçbirim soneklerini getirmelisiniz.

3.4. İlgili Tanımları Miras Alma

İki uçbirimin tanımları benzer olduğunda, eşdeğerdeki parçaları iki defa vermek gerekmez. **tc** yeteneği ile bunlardan biri diğerinin kurallarını kullanabilir. Böyle bir durumda bir tanım diğerine *başvuruyor* ya da onun yeteneklerini *miras alıyor* deriz.

tc yeteneği uçbirim tanımında tanımlanmış son yetenek olmalıdır. Değeri ise yeteneklerin miras alınacağı uçbirim tanımının ismi olmalıdır. Örnek:

```
N9|aaa|ambassador|aaa-30|ann arbor ambassador/30 lines:\
:ti=\E[2J\E[30;0;0;30p:\
:te=\E[60;0;0;30p\E[30;1H\E[J:\
:li#30:tc=aaa-unk:
```

Burada, **aaa-30** uçbirim türü (ayrıca düz **aaa** olarak da bilinir), herşeyin ekran yüksekliğine bağımlı olarak tanımlandığı Ambassador **aaa-unk** tanımının kurallarıyla tanımlanmaktadır. **aaa-36**, **aaa-48** ve benzerleri farklı ekran yükseklikleri sağlamak için benzer şekilde **aaa-unk** tanımını miras alırlar.

aaa-30 tanımının içerdiği **li** (tanımın kaç satırlık olacağını belirler), **ti** ve **te** (uçbirimi bu satır sayısına göre yapılandırır) yetenekleri, miras alınan aynı yetenekleri değiştirmek içindir.

aaa türü etkin uçbirim tanımı yukarıdaki tanıma **aaa-unk** tanımının eklenmiş olur. **tgetent** işlevi **tc** yeteneğini gördüğünde ek *uçbirim tanımını bulur* (sayfa: 5) ve ikisini birleştirir. Diğer taraftan, **tc** yeteneğinin kullanımı sadece uçbirim tanımları yapanları ilgilendirir. Kullanıcılar ve uygulama geliştiricilerin bu yetenekle ilgilenmeye ihtiyaçları olmayacaktır.

Miras alınan uçbirim tanımı sona eklendiğinden, miras alan tanım, miras alınan tanımın içindeki yetenekleri tekrar tanımlayarak bu yeteneklerin tanımlarını değiştirebilir.

Ayrıca miras alan tanım, yeteneklere yeni bir değer belirtmeyerek miras aldığı yetenekleri iptal edebilir. Bunun için yetenek isminden sonra bir **@** karakteri konur. Örnek:

```
NZ|aaa-30-nam|ann arbor ambassador/30 lines/no automatic-margins:\n      :am@:tc=aaa-30:
```

3.5. Etkin Veritabanı Değişirse

Her uygulama uçbirim tanımını veritabanından okumalıdır, böylelikle veritabanındaki bir değişikliğin, değişiklik yapıldıktan sonra başlatılan tüm süreçlerde etkin olması sağlanabilir.

Değişiklik, değişiklik yapılmadan önce başlatılmış süreçler için normal olarak etkin olmayacaktır. Uygulama başlatılırken, uçbirim tanımını bir kere okuyacak ve okuduğunu kullanacaktır. Eğer uygulama, veritabanındaki değişiklikleri algılama özelliğine sahip değilse, süreci sonlandırıp uygulamayı yeniden başlatmalısınız.

Eğer kullanılan tanım **TERMCAP** ortam değişkeninden geliyorsa, veritabanı dosyası bundan etkilenir ve veritabanındaki değişiklikler **TERMCAP** değişkenini değiştirmedikçe etkisiz kalır. Örneğin, bazı kullanıcıların **.login** dosyaları, uygulamaların başlatılmasını hızlandırmak için uçbirim tanımını **TERMCAP**'a kopyalar. Eğer bunu yapmışsanız, veritabanındaki değişikliklerin etkili olabilmesi için **TERMCAP** değişkenini değiştirmeniz gerekecektir.

4. Uçbirim Yeteneklerinin Tanımları

Bu kısım çeşitli alt bölümlere bölünmüştür. Her bölümde uçbirim kullanımı ile ilgili ayrı bir özellik ele alınmıştır. Bir uygulama geliştiriyorsanız sadece sizi ilgilendiren alt bölümlere bakmanız yeterlidir, ama yeni bir uçbirim tanımı yazıyorsanız her alt bölümü okumalı ve yetenek tanımlarını orada açıkladığı gibi yazmalısınız.

Ekran komutu olma özelliğindeki dizge değerli yetenekler *sayısal parametreler* (sayfa: 11) gerektirebilir. Böyle yeteneklerin çoğu aslında parametre kullanmaz. Eğer yetenek parametre gerektiriyorsa bu, tanımının başında açıkça belirtilir. Basit durumlarda, parametrelerin tamamı tanımın ilk ya da ikinci sözcüğünde tamamlanmış olur. Örneğin, **rp** yeteneği iki parametre gerektiren bir komuttur ve tanımı şöyledir:

Bir **c** karakterini **n** kere yineleyerek çıktılan dizge değerli yetenek.

Daha karmaşık durumlarda ya da çok sayıda parametre olduğunda her biri ayrı ayrı açıklanır.

Bir yeteneğin önerilmediğinden sözediliyorsa, bu, bu yeteneği kullanan uygulamalar yazmayın demektir, fakat uçbirim tanımı hala o yeteneği içermeye devam edecektir.

Bir yeteneğin atıl olduğundan sözediliyorsa, bu, bu yeteneğin artık uçbirim tanımlarında da bulunmadığı anlamına gelir.

4.1. Temel Karakteristikler

Bu bölüm uçbirimin temel ve doğal özellikleri olan yetenekleri belgeler. Bunlar ayrıca, karakterlerin çıktılanması ile de ilgilidirler.

os

Uçbirimin üste yazmama özelliği olup olmadığını belirleyen seçimlik yetenek. Bu yetenek etkinse, bir karakterin yazıldığı konumda bir karakter varsa, yeni karakter bu karakteri silmez. Üste yazmamanın etkin olduğu uçbirimler, yazdırma uçbirimleri, saklama tüpleri (günümüzde tamamen atıldılar) ve bazı biteşlemler ekranlardır.

eo

Uçbirimde üste yazmama özelliği etkin olsa bile çıktılanan bir boşluk karakterinin aynı konumdaki bir karakteri silmesinin mümkün olup olmadığını belirleyen seçimlik yetenek. Eğer bu yetenek etkin değilse ve üste yazmama yeteneği etkinse, çıktılanan bir boşluğun imleci hareket ettirmekten başka bir etkisi olmaz.

(Üste yazmamanın kapalı olduğu uçbirimlerde, çıktılanan bir boşluğun aynı konumdaki bir karakteri daima sileceğini varsayabilirsiniz.)

gn

Uçbirimin gerçek bir uçbirim olmayan bir sosyal tür olup olmadığını belirten seçimlik yetenek. Sosyal uçbirim türleri, kullanıcının sisteme bağlanırken atanan öntanımlı uçbirim türü olduğu ve kullanıcının bağlandıktan sonra gerçek uçbirimi kendisinin seçeceği düşünülerek tasarlanmışlardır. Sosyal uçbirim türüne bir örnek **network** türüdür.

Eğer kullanıcı bir sosyal uçbirimi gerçek bir uçbirim ile değiştirememişse, sosyal türler bir uçbirimin yerine getirmesi gerekenlerin nasıl yapılacağını belirtmediğinden, termcap kullanan uygulamalar uçbirimi destek açısından daima çok yetersiz bulacaklardır. **gn** yeteneği bu uygulamaları farklı bir hata iletişi ile kullanıcılarını uyarmak için yönlendirir: "Uçbiriminiz çok yetersiz" demek yerine "Gerçek bir uçbirim türü belirtmemişsiniz." diye uyarı verirler.

hc

Uçbirimin yazılımsal değilde donanımsal (yazıcı gibi) olduğunu belirten seçimlik yetenek.

rp

Bir *c* karakterini *n* kere yineleyerek çıktılanan dizge değerli yetenek. İlk parametre yinelenecek karakterin ASCII kodu, ikinci parametre yineleme sayısıdır. Bu komut çoğunlukla karakterin yineleme sayısı ile orantılı bir geciktirme gerektirir. Bu etki, % dizili parametre aritmetiği kullanarak hesaplanan değeri komut dizgesinin önüne yerleştirerek sağlanabilir, bu durumda **tputs** onu geciktirme miktarı olarak ele alacaktır.

hz

Bu seçimlik yetenek, varlığında uçbirimin ~ karakterini bir ekran komutu olarak kullanılması nedeniyle çıktılanmayacağını belirtir.

Bu seçeneğin varlığında uygulamalar çıktılanacak tüm metni tarayıp ~ karakterlerini başka karakterlerle değiştirmelidir. Bu yapılmadığında ekran görüntüsü bozunmaya uğrayabilir.

Günümüzde nadiren rastlanan eski Hazeltine uçbirimler böyleydi. Bu bakımdan bu seçeneği desteklemek konusunda tasalanmanız gerekmez.

cc

Uçbirimde komut karakterinin öntanımlı komut karakterinden farklı bir karakterle temsil edilmesi durumunda bu karakteri belirtmek için kullanılan dizge değerli yetenek. Öntanımlı komut karakteri **<ESC>**'dir.

Uçbirim tanımındaki tüm komut dizgeleri öntanımlı komut karakteri kullanılarak yazılmalıdır. Eğer komut karakterini değiştiren bir uygulama yazıyorsanız, tüm ekran komutlarının yeni komut karakterini nasıl kullanacaklarını belirtmek için **CC** yeteneğini kullanın.

Çoğu uygulamalar **CC** yeteneğine bakma gereği duymazlar.

xb

<ESC> ve <Control>—<C> karakterlerini gönderemeyen Superbee uçbirim türünü tanımlayan seçimlik yetenek. Bu yeteneği destekleyen uygulamalar <F1> ve <F2> tuşlarından gönderilen kod dizilerini aldıklarında sırayla <ESC> veya <Control>—<C> basılmış gibi işlem yapar. Fakat bu yeteneğin kullanılması önerilmemekte ve artık desteklenmemektedir.

4.2. Ekran Boyutları

Bir uçbirimin sütun ve satır sayısını belirleyen **co** ve **li** isimli iki yeteneği vardır. Fakat ekran boyutları denince bundan fazlası bilinmek istenir.

Bazı işletim sistemlerinde "ekran" aslında bir penceredir ve etkin genişliği değişkendir. Bu sistemlerin bazılarında, uçbirim tanımında ne yazdığına bakmadan **co** yeteneğinden dönecek değere karar vermek için **tgetnum** kullanılır. Diğerlerinde ise uygulamalar pencere genişliğini öğrenmek için bir sistem çağrısı yaparlar. Örneğin BSD 4.3 sistemlerinde, o anki ekran boyutunu öğrenmek için **ioctl** işlevi **TIOCGWINSZ** ile çağrılır.

Tüm pencereli sistemlerde kullanıcı pencerenin boyutlarını değiştirdiğinde uygulamayı uyarmak bakımından termcap yetersiz kalır. Uygulama yazılımları bu olasılığı sisteme bağlı olarak değerlendirmelidir. Bazı sistemlerde C kabuğu pencere değişikliklerini algılayıp **TERMCAP** ortam değişkenine uygun değeri atayarak bu sorunu çözer. Bu sadece uygulamaların ilk başlatılışlarında uygulamalara yardımcı olur, çalışmakta olan uygulamalara değil.

BSD 4.3 dahil bazı sistemlerde ise, ekran boyutları değiştiğinde bir uçbirim kullanan tüm uygulamalar **SIGWINCH** isminde bir sinyal alırlar. Termcap kullanan uygulamalar bu sinyali aldıklarında, yeni ekran boyutlarını öğrenmek için **ioctl** **TIOCGWINSZ** çağrısı yapmalıdır.

co

Karakter sayısı cinsinden ekran genişliğini veren sayısal değerli yetenek. Yazıcı türü uçbirimlerin bile bir **co** yeteneği vardır.

li

Ekranın satır sayısı cinsinden yüksekliğini veren sayısal değerli yetenek.

4.3. İmleç Hareketleri

Termcap, uçbirimin görünür bir işaretleyici olarak bir **imleç**'e sahip olduğunu ve ekran komutlarının bu imlecin konumuyla etkileşim içinde olacağını varsayar. İmlecin belirtilen bir konuma taşınmasının izlenmesi çok önemlidir.

Farklı imleç hareketlerini yerine getiren bir çok uçbirim yeteneği vardır. Çoğu uygulama bunların hepsini kullanmayacak olsa da bir uçbirim tanımı bunların çoğunu tanımlamalıdır. **cm** isimli bir yetenek imleci ekran üzerinde keyfi bir konuma taşır; her uygulama için bunun olması yeterli olduğu halde, sadece görelî hareket komutlarına sahip bazı eski ve zayıf ekranlar ve yazıcı türü uçbirimler bunu destekleme gereği duymazlar. Diğer imleç hareket komutları eniyileme amaçlıdır ve uygulamaların bazı durumlarda birkaç karakter çıktılmasını mümkün kılarlar.

Uygulamanızda görelî imleç hareket komutlarını kullanmayı düşünüyorsanız, imlecin başlangıç konumunu bilmeniz gerekir. Bunu yapmak için, imleç konumlarını takip etmeli ve uçbirime yapılan her çıktılama (karakter

çıkılama dahil) kayıtlarınızı güncellemelisiniz. Buna ek olarak uçbirimin imleci sağ kenardan alt satırın başına döndürüp döndürmediğini de saptamanız gerekir. Bkz. *Sarmalama* (sayfa: 25).

Üzerinde durmamız gereken bir hareket yeteneği de **nw** yeteneğidir. Bu yetenek imleci alt satırın başına taşır. Bunu yaparken bazan başına getirildiği satırı temizler bazan da hiçbir temizlik yapmaz. Bu yetenek, **cm** veya **do** gibi başka hareket yetenekleri olmayan uçbirimlerde bile olması gereken ortak yeteneklerden biridir. Yazıcı türü uçbirimler bile bu yeteneği destekler.

cm

İmleci *l*'inci satırın *c*'inci sütununa konumlandırın dizge değerli yetenek. Her iki parametrenin başlangıç konumları sıfırdır ve ekran belleğine değil görünen ekrana görelidir.

Birkaç çok eski ve atıl olmuş uçbirim dışında tüm uçbirimler **cm** yeteneğini destekler; bu bakımdan, **cm** yeteneğinin yokluğu uygulamalar açısından kabul edilebilir değildir.

ho

İmleci ekranın sol üst köşesine (bu konuma **ev konumu** denir) konumlandırın dizge değerli yetenek. Ekranın sol üst köşesi ekran belleğinin başlangıcı değilse, bu komut imleci görünür ekranın sol üst köşesine konumlandırır, ekran belleğinin başlangıcına değil.

Ekran uçbirimlerinin hepsi bu yeteneği destekler ve bu yeteneğin yokluğu uygulamalar açısından kabul edilebilir değildir.

ll

İmleci ekranın sol alt köşesine konumlandırın dizge değerli yetenek. Bazı uçbirimler bunu ev konumundan taşıyarak yaparlar, ancak uygulamalar bu tür bir işlemi asla varsaymamalıdır. Sadece **ll** dizgesini çıkıtılmalıdır (eğer varsa); önce ev konumuna gidip bundan sonra sol alt köşeye gitmenin en iyi yolu budur ve **ll** komutu bunu yapacaktır.

cr

İmleci üzerinde bulunduğu satırın başına taşıyan dizge değerli yetenek. Uçbirimde bu yetenek yoksa, uygulamalar bu işlemi ASCII satırbaşı karakterini kullanarak yapabilir.

le

İmleci bir sütun sola taşıyan dizge değerli yetenek. Seçimlik **bw** yeteneği belirtilmedikçe, eğer imleç satırın başındaysa, bu komutun etkisi belirsiz olacağından komutu böyle bir durumda kullanmayın. Eğer **bw** belirtilmişse, bu komut imlecin satırbaşında olduğu durumda da kullanılabilir; bu takdirde imleç önceki satırın son sütununa taşınır.

nd

İmleci bir sütun sağa taşıyan dizge değerli yetenek. İmleç satırın sonundaysa bu komutun etkisi belirsiz olacaktır; **am** seçeneği mevcut olmasa bile bu komutu burada kullanmayın.

up

İmleci bir satır üste taşıyan dizge değerli yetenek. İmleç zaten en üst satırdaysa komutun etkisi belirsizdir; bu tarz kullanılmamalıdır.

do

İmleci bir satır alta taşıyan dizge değerli yetenek. İmleç zaten en alt satırdaysa komutun etkisi belirsizdir; bu tarz kullanılmamalıdır.

Bazı uygulamalar, imlecin son satırda olması durumunda, eğer **sf** tanımlanmış ama **sr** tanımlanmamışsa, **do** kullanarak bir satırlık yukarı kaydırma yaparlar. (Bu sadece, eski ve/veya davranışları yanlış olan uçbirimlerde durumu kurtarmak için yapılırsa da yaygın olarak kullanıldığı görülmektedir.) Fakat, doğru çözüm, eğer komut **do** kullanarak kaydırmayla uygulanacaksa uçbirim tanımında **sf** yeteneğini tanımlamaktır.

Özgün fikir, bu dizgenin bir satırsonu karakteri içermeyeceği şeklindedir, dolayısıyla komut, çekirdeğin genel alışkanlığı olan satırsonu karakterinin bir CRLF ile değiştirilmesi iptal edilmeksizin kullanılmalıdır. Ancak, bazı uçbirim tanımlarında **do** dizgesi bir satırsonu içerebilmektedir; böyle bir durumda uygulama çekirdeğin çıktı dönüşümünü iptal etmelidir (bkz. [Termcap Kullanımı için İklendirme](#) (sayfa: 8)).

bw

Varlığında, **le** komutunun sıfırıncı sütunda kullanıldığında imlecin önceki satırın son sütununa taşınacağını belirten seçimlik yetenek. Bu seçeneğin yokluğunda, **le** komutu sıfırıncı sütunda kullanılmamalıdır.

nw

Büyük olasılıkla taşımadan önce satırın kalanını silerek, imleci sonraki satırın başına taşıyan dizge değerli yetenek.

DO, UP, LE, RI

İmleci *n* satır aşağı veya yukarı ya da *n* sütun sola veya sağa taşıyan dizge değerli yetenekler. Bu komutları, imleci ekranın kenarlarından dışarı taşıma anlamında kullanırsanız, etkileri belirsizdir. Sadece birkaç uçbirim tanımı bu komutları içerir ve çoğu uygulama bu komutları kullanmaz.

CM

İmleci, ekran belleğine göre olarak, *l*'inci satırın *c*'inci sütununa taşıyan dizge değerli yetenek. Her iki parametre içine başlangıç sıfırdır. Bu yetenek, ekrana göre adreslemesi belleğe göre adreslemesinden farklı olan uçbirimlerde vardır ve bu tür uçbirimlerin hepsinde de yoktur.

ch

İmleci aynı satırın *c*'inci sütununa taşıyan dizge değerli yetenek. Bu, **cm** yeteneğinin düşey konumu değiştirmeyen özel bir durumudur. **ch** yeteneği, eğer bu özel durumda çıktı **cm** yeteneğinden elde edilenden daha hızlı elde edilebiliyorsa uçbirim tanımına konur. Uygulamalar, çoğu uçbirimin bu yeteneği içermediğini varsaymalıdır.

cv

İmleci aynı sütunda *l*'inci satıra taşıyan dizge değerli yetenek. Bu, **cm** yeteneğinin yatay konumu değiştirmeyen özel bir durumudur. **ch** yeteneği, eğer bu özel durumda çıktı **cm** yeteneğinden elde edilenden daha hızlı elde edilebiliyorsa uçbirim tanımına konur. Uygulamalar, çoğu uçbirimin bu yeteneği içermediğini varsaymalıdır.

sc

Uçbirimin kaydettiği imleç konumunu almak için dizge değerli yetenek. Sadece son kaydedilen konum kullanılabilir. Eğer bu yetenek varsa, ayrıca **rc** yeteneğinin de olması gerekir. Çoğu uçbirim ikisini birden sağlamaz.

rc

İmleci son kaydedilen imleç konumuna taşıyan dizge değerli yetenek. Eğer bu yetenek varsa, ayrıca **sc** yeteneğinin de olması gerekir. Çoğu uçbirim ikisini birden sağlamaz.

ff

Yazıcı türü uçbirimlerde sonraki sayfaya geçmeyi sağlayan dizge değerli yetenek.

ta

İmleci sağdaki sonraki donanımsal sekme konumuna taşıyan dizge değerli yetenek. Eğer uçbirim donanımsal sekme konumlarına sahip değilse bu yetenek yoktur. Çekirdek, sekmelerin boşluklarla doldurma yöntemi ile kullanıldığı uçbirim kiplerine sahipse bu yeteneği kullanmayın.

bt

İmleci soldaki sonraki donanımsal sekme konumuna taşıyan dizge değerli yetenek. Eğer uçbirim donanımsal sekme konumlarına sahip değilse bu yetenek yoktur. Çekirdek, sekmelerin boşluklarla doldurma yöntemi ile kullanıldığı uçbirim kiplerine sahipse bu yeteneği kullanmayın.

Aşağıdaki yetenekler uçbirimlerin tanımlarında vardır ama yeni uygulamalarda kullanılması önerilmemektedir.

nc

Varlığı, uçbirim **cr** yeteneği olarak ASCII satırbaşı karakterini desteklemiyor anlamına gelen seçimlik yetenek. Eski uygulamalar **cr** yeteneğinin yokluğunda bu yeteneğin olduğunu varsaydıkları için bu yetenek hala vardır. Satırsonu karakterinin kullanılamayabileceğini eski uygulamalara belirtmek için **nc** yeteneği kullanılır.

Yeni uygulamalar öntanımlı bir **cr** yeteneğinin varolacağını varsaymamalı ve **nc** yeteneğinin varlığına bakmamalıdır. Yine de, uçbirim tanımları **cr** yeteneğini içermiyorsa, **nc** yeteneğini içermelidir.

xt

Varlığı, uçbirimin imleç konumlamada ASCII sekme karakterini kullanmayabileceği anlamına gelen seçimlik yetenek. Eski uygulamalar **ta** yeteneğinin yokluğunda bu amaçla ASCII sekme karakterinin kullanılabilceğini varsaydıklarından bu yetenek vardır. Eski uygulamalara sekme kullanmayın diyebilmek için **xt** yeteneğini kullanıyoruz.

Yeni uygulamalar öntanımlı bir **ta** yeteneğinin varolacağını varsaymamalı ve **xt** yeteneğinin varlığına bakmamalıdır. Ayrıca, **xt** yeteneğinin standart çıktı kipiyle ilgili bazı sorunlarının olduğu gözardı edilmemelidir (bkz. *Çarpıcı ve Sıradan Görünüm Kipleri* (sayfa: 33)). Yeteneğin standart çıktıyla ilgili sorunlar sebebiyle değil imlecin konumlanması ile ilgili olarak kullanılmaması önerilmektedir.

Aslında, **xt** yeteneği uçbirimin bir Teleray 1061 olduğu anlamına gelir.

bc

le yeteneğine eşdeğer tamamen atıl olmuş bir yetenek.

bs

Varlığı, imleci sola hareket ettirmek için ASCII gerisilme karakterinin kullanılabilceğini belirten seçimlik yetenek. Atıl bir yetenektir; bunun yerine **le** yeteneğinin varlığına bakın.

nl

İmleci duruma göre alt satıra taşıyan ya da kaydırma yapan, ancak kullanılması önerilmeyen dizge değerli yetenek. Aynı komut, imleç en alt satırda ise ekranı kaydırmalı, başka bir satırda ise alt satıra geçmelidir. Yeni uygulamalar **do** veya **sf** kullanmalı, **nl** yeteneğini yoksaymalıdır.

nl yeteneğinin yokluğunda, eski uygulamalar bu amaçla satırsonu karakterlerinin kullanılabilceğini varsayar. Bu uygulamaların iyi bir seçim yapmadıkları söylenebilir de, eğer bir alt satıra geçmenin en iyi yolu bir satırsonu karakteri kullanmaktır demek istemiyorsak, bir uçbirim tanımında **nl** yeteneğinin tanımlı olması hala arzu edilebilir olacaktır.

4.4. Sarmalama

Sarmalama imleci sağ kenardan sonraki satırda sol kenara götürmek demektir. Bazı uçbirimler bir karakter sağ kenarda çıktığında bunu özdevinimli yaparken bazıları bunu yapmaz. Termcap kullanan çoğu uygulama uçbirimin sarmalama yapıp yapmadığını bilmek ister. Bunun olup olmadığını belirleyen iki seçimlik yetenek vardır.

am

Son sütuna yazılan karakterin imlecin sonraki satırın başına taşınmasına sebep olacağını belirten seçimlik yetenek.

am yeteneği yoksa, bir karakterin son sütuna yazılması imleci yerinde bırakır.

am yetenekli bir uçbirimde son satırın son sütununa yazmaktan kaçınılmalıdır. Aksi takdirde, bu bir kaydırma ile sonuçlanabilir (bkz. *Kaydırma* (sayfa: 25)) ve bu istenen birşey değildir.

Yazılımınız **am** seçeneğinin varlığına bakmayı gerektiriyorsa, sarmalamanın tuhaf bir yolla yapılacağını belirten **xn** seçeneğinin varlığına da bakması gerekir. Bilinen çoğu uçbirim **xn** seçeneğine sahiptir.

xn

İmlecin tuhaf bir yolla sarmalanacağını belirten seçimlik yetenek. Tuhaf davranışın bilinen en azından iki çeşidi vardır. Termcap veritabanı bu ikisini ayrımsamayı sağlayacak hiçbir şey içermez.

Concept-100 uçbirimlerde, son sütuna yapılan çıktılama sonucunda sarmalama hemen hemen sıradan bir **am** uçbirimdeki gibidir. Fakat bu son karakter bir satırsonu karakteri ise karakter yoksayılar.

DEC VT-100 uçbirimler (sarmalama seçeneği etkinse) başka bir tuhaf şey yapar: imlecin sarmalanması için son sütuna yazılacak karakterden sonra bir karakter gelmesi gerekir. Aslında sarmalama, son sütuna yazılacak karakter yazılmadan önce sonraki karakterin alınmasıyla gerçekleşir.

Bu uçbirimlerin ikisinde de, son satıra bir karakter yazıldıktan hemen sonra alt satırın başına sonraki karakter yazılır. Fakat satırsonu, gerisilme gibi görelî imleç hareket karakterlerinin etkisi uçbirime bağlı olarak değişiklik gösterir. Ayrıca silme ve kaydırma komutlarının da etkileri uçbirime bağlıdır. **xn** yeteneğine sahip bir uçbirimin neler yapacağı konusunda hiçbir önkabul yapamazsınız. Dolayısıyla, ne yaptığınızdan emin olmak istiyorsanız, bu tür uçbirimlerde sarmalama işlemini uçbirimin insafına terketmemelisiniz.

xn yeteneğine sahip bir uçbirimde kararlı sonuçlar almak istiyorsanız, son sütuna yazdıktan sonra, bir **cm** mutlak konumlama komutu kullanın. Diğer bir kararlı sonuç çözümü, imleci sonraki satırın başına konumlandırılan satırbaşı-satırsonu karakter çiftini çıktılamaktır.

LP

Varlığı halinde, son satırın son sütununa yapılan bir yazma işleminin istenmeyen bir kaydırmaya yol açmayacağını garanti eden seçimlik yetenek. **LP** DEC'in **xn** tuhafliklarının belirtisidir.

4.5. Kaydırma

Kaydırma denince, ekran içeriğinin bir satır yukarı ya da aşağı kaydırılmasını anlayacağız. İçeriğin yukarı kaydırılmasına **ileri kaydırma**, aşağı kaydırılmasına ise **geri kaydırma** diyeceğiz.

Çoğu uçbirimde sıradan çıktılama sırasında her satırın çıktılanmasından sonra kaydırma yapılır. Fakat, rasgele erişimli çıktılama için termcap kullanan uygulamalar, ekran içeriğini kaydırmak gerektiğinde bu bölümde açıklanan komutları kullanırlar.

Bazı uçbirimler **kaydırma bölgesi** denen bir özelliğe sahiptir. Bu özellik, kaydırmanın etkisini belli sayıda satırla sınırlayabilmenizi sağlar. Bu sayıda satır dışında satır kaydırmadan etkilenmez. Kaydırma bölgesi özelliğinin varlığını gösteren yetenekler **cs** veya **cS** yetenekleridir.

sf

İmlecin son satırın başlangıcında olduğu kabulüyle, ekran içeriğini bir satır yukarı kaydıran dizge değerli yetenek.

sr

İmlecin ilk satırın başlangıcında olduğu kabulüyle, ekran içeriğini bir satır aşağı kaydıran dizge değerli yetenek.

do

Bir kaç uygulama **sf** yeteneğinin yaptığını yapmak için **do** kullanmayı dener. Bu aslında doğru bir öngörü değildir; bu yetenek bazı eski uçbirim tanımlarında bir **sf** komutunun yokluğunu gidermek için vardır.

Bu uçbirim tanımları **sr** tanımı içerdiklerinden, bir zamanlar **do** tanımı da farklıydı ve kaydırma amacıyla da kullanılırdı. Fakat, kaydırma işlemi imleci bir satır aşağı taşımaktan daha fazla geciktirme gerektirdiğinden, bu iki işlevin tek bir komutta birleştirilmesi arzu edilen bir şey değildir. **sf** ve **do** tanımlamasının ayrı ayrı yapılması geciktirmenin gerektiği gibi yapılmasını mümkün kılar. Ayrıca, tüm kaynaklar, kaydırma konusunda **do** yeteneğine bel bağlanmaması gerektiğinde hem fikirdir

Bu durumda en iyi yaklaşım, bu uçbirim tanımlarına **sf** yeteneğinin eklenmesi, eğer kaydırma yapılıyorsa, **do** tanımının kopyalanmasıdır.

SF

İmlecin son satırın başlangıcında olduğu kabulüyle, ekran içeriğini *n* satır yukarı kaydıran dizge değerli yetenek.

SR

İmlecin ilk satırın başlangıcında olduğu kabulüyle, ekran içeriğini *n* satır aşağı kaydıran dizge değerli yetenek.

cs

Kaydırma bölgesinin belirtilebilmesini sağlayan dizge değerli yetenek. Bu komut iki parametre alır: kaydırma bölgesinin başlangıcının (ilk satır için sıfırdır) belirtildiği *başlangıç* parametresi ile son satırının belirtildiği *son* parametresi. Bir kaydırma bölgesi belirtildiğinde, kaydırma işlemi belirtilen satır aralığı ile sınırlanır; aralığın dışında kalan satırlar kaydırma işleminden etkilenmez.

İmleci kaydırma bölgesinin dışına taşımaya çalışmayın. Bölge, açıkça kaldırılana kadar belirlenmiş olarak kalır. Kaydırma bölgesini kaldırmak için, kaydırma bölgesi olarak tam ekran yüksekliği belirtilmiş başka bir **cs** komutu kullanılır.

cs komutuyla kaydırma bölgesi tanımlandığında, imleç konumu tanımsız olur, dolayısıyla hemen ardından imleci **cm** ile konumlamak gerekir.

cs

Kaydırma bölgesinin belirtilebilmesini parametreleri farklı bir biçimde kullanarak sağlayan dizge değerli yetenek. Etkisi **cs** ile aynıdır. Farklı olarak dört parametre gerektirir:

1. Ekrandaki toplam satır sayısı.
2. Kaydırılmak istenen bölgenin yukarısındaki satır sayısı.
3. Kaydırılmak istenen bölgenin aşağısındaki (bölgenin dışındaki) satır sayısı.
4. İlk parametre ile aynı olarak ekrandaki satır sayısı.

Bu yetenek Ann Arbor Ambassador'un kaydırma bölgesi özelliği için tasarlanmış bir GNU oluşumdur; istenirse, **cs** dizgesine Unix dışı %'li diziler konularak da yapılabilir, fakat Unix termcap'inin **cs** yeteneğini kullanan Unix uygulamalarında sorun çıkabilirdi. Şimdilik **cs** yeteneğini sadece GNU Emacs kullanmaktadır.

ns

Varlığında, uçbirimin sıradan çıktığı normalde kaydırmayacağını belirten seçimlik yetenek. Günümüz uçbirimlerinde, bu, son satırın son sütununda çıktılanan bir satırsonu karakteriyle imlecin ilk satırın başına taşınacağı anlamına gelir. Bazı atıl uçbirimlerde farklı şeyler olabilir.

Bu yeteneğe sahip olsa bile uçbirim kaydırma yapabilir. Eğer **sf** yeteneği mevcutsa, **ns**'in varlığına rağmen kaydırma için kullanılabilir.

da

Varlığında, satırların yukarı kaydırılmasında ilk satır aşıldığında, eğer ardından aşağı kaydırma yapılırsa ekranın geri gelebileceğini belirten seçimlik yetenek.

da ve **db** seçenekleri kaydırmanın yapılışını kesinlikle etkilemez. Fakat, kaydırma yapan uygulamalar, eğer bu seçenekler uçbirim tanımında varsa, ekranda kaydırılan satırları temizlemek isterler.

db

Varlığında, satırların aşağı kaydırılmasında son satır aşıldığında, eğer ardından yukarı kaydırma yapılırsa ekranın geri gelebileceğini belirten seçimlik yetenek.

lm

Uçbirimin, ekran belleğinde sahip olduğu satır sayısını belirten sayısal değerli yetenek. Sıfır değeri, uçbirimin bir ekranlık ekran belleğinden fazlasına sahip olduğu anlamına gelir, ama satır sayısı sabit değildir. (Satır sayısı her satırdaki metnin miktarına bağlı olabilir.)

SF yeteneğinin tanımlı olduğu her uçbirim ayrıca, **sf** yeteneğini de tanımlamalıdır; aynı şekilde **SR** tanımlıysa **sr**'de tanımlı olmalıdır. Bununla birlikte, çoğu uçbirim bir kerede sadece bir satırlık kaydırma yapabilir ve bu durumda **sf** yeteneği olduğu halde **SF** yeteneği tanımlanmamış olabilir, Benzer şekilde **SR**'siz **sr** olabilir.

Bu bakımdan, kaydırma oluşumlarını kullanan uygulamalar, **SF** yeteneğinin yokluğunda **sf** ile, benzer şekilde **SR**'nin yokluğunda **sr** ile çalışabilmelidir. Diğer yandan, bazı yavaş uçbirimlerde, **SF** değil sadece **sf** kullanan bir uygulama da olabilir.

Bir kaydırma komutunu **tputs** ile çıktılarken, *satır_sayısı* argümanı ekranın kaydırılmak istenen bölümünün satır sayısı toplamına eşit olmalıdır. Çoğunlukla bu komutlar satır sayısına bağlı geciktirme gerektirirler (bkz. [Geciktirme](#) (sayfa: 8)).

4.6. Pencereler

Termcap'ta bir **pencere**, ekranın dikdörtgen biçiminde bir parçasıdır ve tüm ekran işlemleri bu alanda sınırlanır. Sarmalama, temizleme, kaydırma, ekleme ve silme işlemleri belirtilen pencerede ekranın tamamında uygulandığı gibi uygulanır.

wi

Uçbirimin çıktılama için kullanacağı ekran penceresini ayarlamakta kullanılan dizge değerli yetenek. Hepsinin başlangıcı sıfır olmak üzere bu komut dört parametre alır:

1. Pencerenin ilk satırı olacak satır.
2. Pencerenin son satırı olacak satır.

3. Pencerenin ilk sütunu olacak sütun.
4. Pencerenin son sütunu olacak sütun.

Çoğu uçbirim pencereleri desteklemez.

4.7. Ekran Parçalarının Temizlenmesi

Ekranın çeşitli parçalarını boşaltmak anlamında temizleyen çeşitli uçbirim yetenekleri vardır. Çoğu ekran uçbirimi bu yeteneklerin hepsini desteklerken, **c1** dizgesini bütün ekran uçbirimleri destekler.

c1

Ekranın tamamını temizledikten sonra imleci üst sol köşeye yerleştiren dizge değerli yetenek.

cd

İmlecin üzerinde bulunduğu satırdan itibaren bu satır dahil ekranın alt satırına kadar ekranı temizleyen dizge değerli yetenek. Bu komut sadece imleç sıfırıncı sütundayken kullanılmalıdır; aksi takdirde komutun ne yapacağı belirlenmemiştir.

ce

İmleçten satırın sonunu kadar satırı temizleyen dizge değerli yetenek.

ec

İmlecin üzerinde bulunduğu karakterden başlayarak *n* karakteri temizleyen dizge değerli yetenek. Bu komutun imlecin yerini değiştirmeyeceği umulur. *n* parametresi ile belirtilen karakter sayısı ekranın sağ kenarını aşacak sayıda olmamalıdır; aksi takdirde komutun ne yapacağı belirlenmemiştir.

Satır sonuna kadar temizleme (**ce**) ekranı güncelleyen uygulamalar için hayati önemdedir. Hemen hemen tüm ekran uçbirimleri bu işlemi desteklediğinden, **ce**'nin yokluğunda bir uygulamaya böyle bir uçbirimle çalışmayı reddedebilir. Yine de, bu sınırlamayı koymak istemiyorsanız, satır sonuna kadar temizleme işlemini sağ kenara kadar boşluk karakterleri çıkıtılarak bunu sağlayabilirsiniz. bunu yaparken imleci yerinde bırakmak için imlecin başlangıç konumunu hatırlamanız gerekir. Ayrıca bu teknik boşluk yazmanın silme olacağı kabulüne dayanır. Fakat bu sadece **ce** desteği olmayan ekran uçbirimleri için doğrudur.

4.8. Satır Ekleme/Silme

Satır eklemek mevcut metin satırları arasında, ekranın ortasında boş bir satır oluşturmaktır. Satırın eklendiği yerin altında kalan satırlar aşağı kayıp bir satır da ekranın altında kaybolurken, üstünde kalan satırlarda bir değişiklik gözlenmez.

Satır silmek ise ekranda satırları yukarı kaydırmak suretiyle bir satırın altındaki satır tarafından örtülerek görünmemesinin sağlanmasıdır. Bu işlemin sonunda ekranın altında yeni bir satır görünecektir. Bu yeni satır genelde boş olur, ancak **db** yeteneği etkin olan sistemlerde bu satır daha önce ekranın altına kaydırılmış veya ekranın altına eklenmiş bir satır olabilir.

Satırların eklenmesi ve silinmesi görüntünün uzatma veya kısaltma anlamında güncellenmesini sağlayan uygulamalar için yararlıdır. Ayrıca metin düzenleyicilerde ekranın kısmen kaydırılması, kesilmesi ve eklenmek istenen satırların yeniden gösterilmesini sağlamak için de yararlıdır.

Birçok uçbirimde imlecin olduğu konumda tek bir satırın eklenmesi ya da silinmesine yarayan komutlar varken bazıları silinecek ya da eklenecek satır sayısını parametre olarak alan tek bir komutla çok sayıda satırın silinmesini ya da eklenmesini sağlayabilir. Bu komutları kullanmadan önce imleç daima sıfırıncı sütuna taşınır.

al

İmlecin bulunduğu satırın öncesine boş bir satır ekleyen dizge değerli yetenek. İmlecin bulunduğu satır ve onun altındaki tüm satırlar aşağı taşınır. Ekrandaki son satır (veya kaydırılan bölgenin son satırı) görünmez olur; bu satır çoğu durumda yok edilir. Eğer **db** yeteneği mevcutsa, yokedilmeyebilir (bkz. [Kaydırma](#) (sayfa: 25)).

Bu komut kullanılmadan önce imleç sol girintinin içine alınmış olmalıdır. Komut imlecin konumunu değiştirmez.

dl

İmlecin bulunduğu satırı silen dizge değerli yetenek. İmlecin bulunduğu satırın altında kalan tüm satırlar yukarı taşınır. Ekranın altında (veya kaydırma bölgesinin sonunda) bir boş satır görünür. Eğer **db** yeteneği mevcutsa, ekranın altında görünen satır boş olmayabilir.

Bu komut kullanılmadan önce imleç sol girintinin içine alınmış olmalıdır. Komut imlecin konumunu değiştirmez.

AL

İmlecin bulunduğu satırın öncesine *n* boş satır ekleyen dizge değerli yetenek. Bu komut *n* kere yinelenen **a1** komutu imiş gibi görünse de işlemi tek bir **a1** komutu kadar hızlı gerçekleştirir.

DL

İmlecin bulunduğu satırdan başlayarak *n* satırı silen dizge değerli yetenek. Bu komut *n* kere yinelenen **dl** komutu imiş gibi görünse de işlemi tek bir **dl** komutu kadar hızlı gerçekleştirir.

AL komutunu içeren her uçbirim tanımı **a1** komutunu da içermek zorundadır; aynı kural **DL** ve **dl** için de benzer şekilde geçerlidir. Diğer yandan, bir uçbirim tanımı tek bir satır ile işlem yapan **a1** ve/veya **dl** komutlarını içerirken **AL** ve/veya **DL** komutlarını içermeyebilir.

Bu bakımdan, satır ekleme ve silme işlemlerini kullanan bir yazılım **AL** komutunun yokluğu durumunda **a1** komutuyla, **DL** yokluğunda ise **dl** ile çalışabilmeye hazır olmalıdır. Bununla birlikte, sadece **a1** ve **dl** komutlarını kullanan, **AL** ve **DL**'ye hiç bakmayan yazılımlar da geliştirilebilir.

Eğer bir uçbirim satır ekleme ve silmeyi doğrudan desteklemiyor ama bölgesel kaydırmayı destekliyorsa, aynı etki kaydırma işlemi ile sağlanabilir. Bu bakımdan, satır ekleme ve silme işlemleri yapan yazılımlar böyle bir duruma da hazırlıklı olmalıdır. Satır ekleme ve silme işlemleri için tek yolun bu olduğu VT100 uçbirimlerinin yaygın kullanımı gözönüne alındığında böyle bir durumun gözönüne alınmış olması oldukça önem kazanır.

Kaydırma bölgesi ayarlanabilen uçbirimlerde satırların silinmesi veya eklenmesi kaydırma bölgesinden etkilenir. Birkaç ardışık satırın birkaç satır vasıtasıyla yukarı veya aşağı kaydırılmasının istendiği durumlarda yararlıdır. Bkz. [Kaydırma](#) (sayfa: 25).

Ekranın altına itilen satırlar, eğer uçbirim seçimlik **db** yeteneğine sahipse kaybedilmezler; bunlar ekranın görünmeyen bölgesinin belleğine kaydedilirler. Bu işlem, kaydırma ile satırların ekranın altına itilmesi ile aynıdır. Gerek geri kaydırma gerekse satır silme işlemleri sonunda eğer uçbirim seçimlik **db** yeteneğinden başka seçimlik kaydırma bölgesi yeteneğine de sahipse ekranın altında kaybolan bu satırlar tekrar görünecektir.

Ekleme veya silme komutunu **tputs** ile çıktılarken *satır_sayısı* argümanı, imleç konumundan ekranın (veya kaydırma bölgesinin) altındaki satıra kadar tüm satırları sayısını içermelidir. Bu komutlar çoğunlukla satır sayısına göreli bir geciktirme gerektirirler. Bkz. [Geciktirme](#) (sayfa: 8).

AL ve **DL** için *satır_sayısı* argümanı sadece eklenen veya silinen satırların sayısını değil, işlemde etkilenen tüm satırların sayısını içermelidir. **AL** ile *n* satır eklemenin **a1** ile bir satır eklemek kadar hızlı olabilmesi için bu böyledir.

4.9. Karakter Ekleme/Silme

Karakter eklemek, bir satırın ortasında satırın kalanını sağa iterek boş bir alan oluşturmaktır. Bu işlem sırasında en sağ sütundaki karakter kaybedilir.

Karakter silmek, bir satırın ortasında satırın kalanını sola çekmek suretiyle bir karakterin örtülmesini sağlamaktır. Bu işlem sırasında en sağdaki sütunda bir boş alan ortaya çıkar.

Karakterlerin eklenmesi ve silinmesi görüntünün uzatma veya kısaltma anlamında güncellenmesini sağlayan uygulamalar için yararlıdır. Ayrıca metin düzenleyicilerde düzenlenen bir satırda sonuçların yeniden görüntülenmesi için de yararlıdır.

Birçok uçbirimde imlecin olduğu konumda tek bir karakterin eklenmesi ya da silinmesine yarayan komutlar varken bazıları silinecek ya da eklenecek karakter sayısını parametre olarak alan tek bir komutla çok sayıda karakterin silinmesini ya da eklenmesini sağlayabilir.

Çoğu uçbirim bir karakterin belli bir konumda çıktılanması işleminin normalde araya ekleme şeklinde olmasını sağlar. Araya ekleme kipine giriş ve çıkış için özel bir komut dizgesi kullanılır. Bir uçbirimin bir ekleme komutu yerine bir ekleme kipi ile tasarlanma sebebi, karakterlerin eklenmesi işleminin genellikle karakterlerin yazılması sırasında olagelmesidir. Ekleme kipinde bu işlem, kipe giriş ve çıkış için harcanan sabit zaman haricinde karakterlerin normalde yazımı kadar hızlıdır. Hatta, hat hızı yeterinden yüksekse ekleme kipinde karakterlerin çıktılanması için geciktirme bile gerekebilir.

Ekleme kipine girmenizi gerektiren bazı uçbirimler ekleme yapılan her konum için özel bir komut kullanmanızı gerektirebildikleri gibi bazıları da eklenen her karakterden önce ve sonra özel birer komut kullanmanızı da gerektirebilirler.

Karakterlerin silinmesi genellikle bir ya da daha fazla konum için bir komutun doğrudan kullanılması ile gerçekleşir; ancak bazı uçbirimlerde silme komutunu kullanabilmek için özel bir silme kipine girmeyi ve sonrasında da çıkmayı gerektirebilir. Bazan ekleme kipi ile silme kipi aynı kip olur.

Bazı uçbirimler çıktılanmış bir boşluk karakterinin konumu ile karakter içermeyen bir konumu farklı değerlendirir. Böyle uçbirimlerde, karakter ekleme veya silme işlemi etkisini satırın son sütunundaki karakterde değil imleç konumundaki karakterden itibaren gösterir. Hatta ilk satırda kaymayı durduracak bir konum olmadıkça etki birden fazla satırı da kapsayabilir. Bu tür uçbirimler seçimsizlik **in** yeteneğinin varlığı ile tanınırlar.

in yetenekli uçbirimlerde, karakterlerin silinip ardından tekrar yazılması olarak bildiğiniz karakter atlama tekniği daha sonraki karakter silme ve ekleme işlemlerinin bu satır üzerinde standart dışı şeyler yapmalarına sebep olur. Bunun olma ihtimali olan bir programda **in** yeteneğinin varlığına bakmalı ve **in** yeteneğinin varlığı durumunda, boşluk olarak görünecek sütunlara boşluk karakteri yazmaya dikkat etmelisiniz.

Uçbirim yeteneklerinin bolluğu tüm bu karmaşıklığın açıklanmasını gerektirir. Burada hepsini listeledik. Aşağıdaki listeyi yazılımcının bu yetenekleri bir algoritmada hesaba katacağı sıraya göre hazırladık.

im

Ekleme kipine girilmesini sağlayan dizge değerli yetenek.

vi metin düzenleyicisi bir karakterin eklenmesinin sadece **im** yeteneğinin varlığı halinde mümkün olacağını varsaydığından eğer uçbirimin özel bir ekleme kipi yoksa ama karakter eklemek için özel bir komutu varsa uçbirim tanımında **im** yeteneği boş değerle belirtilmiş olmalıdır.

Yeni uygulamalar **vi** gibi davranmamalıdır. **im**'e sadece tanımlanmış olduğu kadar değer biçmeliler.

ei

Ekleme kipini bırakmayı sağlayan dizge değerli yetenek. **im** varsa bu yetenek de olmalıdır.

Ekleme kipine giriş ve çıkış için aynı komutu kullanan sadece bir kaç eski uçbirim kalmıştır ve bu dizge bu uçbirimlerde ekleme kipi etkinse etkisiz, etkisizse etkin yapar. Bu uçbirimler için bundan dolayı **ei** yeteneğinin **im** yeteneğine eşdeğer olduğundan bahsedilebilir. Bu tür uçbirimleri destekleyecekseniz, programınız ekleme kipinde olup olmadığını daima hatırlamak zorunda kalacaktır. Bununla birlikte, bu uçbirimlerin artık kullanılmadığını gözönünde tutarak onları desteklemeyi reddedebilirsiniz. Günümüz uçbirimlerinin hepsinde ekleme kipinden her çıkmak isteyişinizde **ei** komutunu rahatça kullanabilirsiniz.

ic

İmleç konumuna bir karakter ekleyen dizge değerli yetenek. İşlem imleci hareket ettirmez.

Bir karakterin ekleme kipinde çıktılanması için karakteri eklemek yeterliyse, **ic** yeteneği boş değerle tanımlanmalıdır.

Eğer uçbiriminiz ekleme için bir seçim yolu (ekleme kipi ya da özel bir komut olarak) tanımlıyorsa ve **im** tanımlıyken **ic**'nin tanımsız olması çok sayıda karakter eklendiğinde en verimli işlemi belirttiğinden **ic** olmaksızın **im**'i tanımlamayın. Her ekleme yapılışında *her ikisini de* kullanmak gerekir anlamında her iki dizgeyi de *tanımlamayın*.

ip

Ekleme kipinde eklenen karakterden sonrasını çıktıl原因an dizge değerli yetenek. Çoğunlukla bir karakter eklenmesinden sonra gecikme gerekiyorsa geciktirme uygulamak için kullanılır (bkz. [Geciktirme](#) (sayfa: 8)).

IC

İmleç konumundan itibaren n karakter ekleyen dizge değerli yetenek. **ic** dizgesinin bir boşlukla n kere yinelenmesine eşdeğerdir.

IC yeteneği varsa uygulamalar ekleme kipine girmeksizin bu yeteneği kullanabilirler.

mi

Varlığının, ekleme kipindeyken imleç taşımanın ve uçbirimin ekleme kipinde kalacağıının varsayılmasının güvenli olduğu anlamına gelen seçimlik yetenek.

in

Varlığı halinde, uçbirimin çıktılanmış bir boşluk karakterinin konumu ile karakter içermeyen bir konumu farklı değerlendireceğini belirten seçimlik yetenek.

Bir uygulama yazılımında uçbirimin karakter eklemesi yapabileceğini varsaymak için **IC**, **im**, **ic** veya **ip** yeteneklerinden birinin tanımlanmış olması yeterlidir.

n boşluk karakteri ekleyip imleci konumlamak için aşağıdaki algoritmayı izleyin:

1. **IC** yeteneği sağlanmışsa dizgeyi n parametrelili çıktılanmak yeterlidir. Aksi takdirde (veya **IC** dizgesine bakmak istemiyorsanız) aşağıdaki adımları uygulayın.
2. Uçbirim zaten ekleme kipinde değilse ve varsa **im** dizgesini çıktıl原因an.
3. 4'den 6'ya kadar ki adımları n kere yineleyin.
4. Varsa **ic** dizgesini çıktıl原因an.
5. Bir boşluk çıktıl原因an.
6. Varsa **ip** dizgesini çıktıl原因an.

7. **ei** çıktılarak ekleme kipinden çıkın. Bunu hemen yapmanıza gerek yoktur. Seçimlik **mi** yeteneği varsa imleci taşıyabilirsiniz ve imleç ekleme kipinde kalır; böylece yeniden ekleme kipine girmeksizin başka ekleme işlemleri de yapabilirsiniz.

Boşluk olmayan n karakter ekleyip imleci konumlamak için aşağıdaki algoritmayı izleyin:

1. Karakterleri çıktılmak için **IC** yeteneği sağlanmışsa dizgeyi n parametrelili çıktılmak yeterlidir. Aksi takdirde (veya **IC** dizgesine bakmak istemiyorsanız) aşağıdaki adımları uygulayın.
2. Uçbirim zaten ekleme kipinde değilse ve varsa **im** dizgesini çıktılın.
3. 4'den 6'ya kadar ki adımları çıktılanacak her karakter için tekrarlayın
4. Varsa **ic** dizgesini çıktılın.
5. Sonraki karakteri çıktılın.
6. Varsa **ip** dizgesini çıktılın.
7. **ei** çıktılarak ekleme kipinden çıkın. Bunu hemen yapmanıza gerek yoktur. Seçimlik **mi** yeteneği varsa imleci taşıyabilirsiniz ve imleç ekleme kipinde kalır; böylece yeniden ekleme kipine girmeksizin başka ekleme işlemleri de yapabilirsiniz.



Bilgi

Bir husus dışında bu, özgün Unix termcap belirtimi ile aynı değildir: belirtim **IC** dizgesinin ekleme kipine girmeksizin kullanılabilmesini varsayar. Bildiğim kadarıyla bu böyledir, ekleme kipine girmek ve çıkmaktan kaçınmak mümkündür ve ayrıca karakterler eklendikleri konumlara gittikten sonra karakter ekleme geciktirmesinden de kaçınılabilir.

Karakterlerin silinmesi bu kadar karmaşık değildir; bir sütunun silinmesi **dc** dizgesi çıktılarak yapılır. Yine de, **dc**'nin işini yapabilmesi için **dm** ile girilmesi gereken bir silme kipi olabilir.

dc

İmleç konumundaki karakteri silen dizge değerli yetenek. Bu yeteneği olmayan bir uçbirim karakterleri silemez.

DC

İmleç konumundan itibaren n karakteri silen dizge değerli yetenek. **dc** dizgesinin n kere yinelenmesi ile aynı etkiye sahiptir. **DC** yeteneği olan her uçbirim **dc** yeteneğine de sahiptir.

dm

Silme kipine girmeyi sağlayan dizge değerli yetenek. Olmayışı silme kipinin olmadığı ve herhangi bir anda **dc** yeteneğinin kullanılabilmesi (**dc** yeteneğinin olduğu varsayımıyla) anlamına gelir.

ed

Silme kipinden çıkmayı sağlayan dizge değerli yetenek. **dm** mevcutsa bu yetenek de olmalıdır.

n karakteri silmek ve imleci konumlamak için bu adımları uygulayın:

1. **DC** yeteneği varsa, parametre n ile kullanmak yeterlidir. Aksi takdirde aşağıdaki adımları uygulamak gerekir.
2. Uçbirim zaten silme kipinde değilse ve varsa **dm** dizgesini çıktılın.

3. **dc** dizgesini *n* kere çıktılaysın.
4. Sonunda **ed** dizgesini çıktılaysın. Eğer seçimlik **mi** yeteneği mevcutsa silme kipine yeniden girmek ve çıkmak gerekmezsiniz imleci taşıyabilir ve başka silme işlemleri de yapabilirsiniz.

IC yeteneğinde olduğu gibi **DC** yeteneğinin de silme kipine girmeksizin çalışacağı (**dc** çalışmasa bile) varsayımıyla özgün termcap belirtiminden ayrılırız.

dm ve **im** yeteneklerinin her ikisi de mevcutsa ve aynı değere sahiplerse bu, uçbirimin hem silme hem de ekleme kipi olarak tek bir kipe sahip olduğu anlamına gelir. Bunun bilinmesi bir program için yararlıdır, çünkü bu durumda silme/ekleme kipini bırakmaksızın ve yeniden girmeksizin silme ve ekleme işlemleri yapılabilir.

4.10. Çarpıcı ve Sıradan Görünüm Kipleri

Görünüm kipleri karakterlerin görünümünü değiştirme yollarıdır. Çok bilinen görünüm kipleri arasında negatif görüntü, karartma, aydınlatma, yanardönerlik, altçizme, görünmezlik ve başka karakter kümesi sayılabilir. Her uçbirim çeşidi bunların hiçbirine veya hepsine ya da bazılarına destek içerebilir.

Her uçbirim türü için, bir metni vurgulu duruma getiren bir görünüm kipi veya bunların birleşimi **çarpıcı görünüm kipi** olarak adlandırılır. Çarpıcı görünüm kipine giriş ve çıkış için **so** ve **se** yetenekleri kullanılır. Görünüm kiplerini sadece bazı metin parçalarını vurgulamak için kullanan programlar genelde çarpıcı görünüm kipini kullanırlar, böylece çok sayıda uçbirim üzerinde çalışabilirler. Altçizme ve diğer karakter kümesi kipi dışında kalan görünüm kiplerinin kullanımları daha az yaygındır.

Görünüm kiplerini gerçekleyen uçbirimler bunu nasıl yaptıklarına bağlı olarak iki genel sınıftan birine girerler.

Bazı uçbirimlerde bir görünüm kipinin varlığı veya yokluğu her karakter konumu için ayrı kaydedilir. Bu uçbirimlerde, yazılan her karakter yazıldıkları andaki görünüm kiplerini de içerir ve bu kipler karakter silininceye ya da üzerine yazılancaya kadar korunurlar. İleride yazılacak karakterler için görünüm kiplerini açıp kapamaya yarayan özel komutlar vardır.

Bazı uçbirimlerde ise görünüm kipinin değiştiğini belirten işaretleyiciler kullanılır ve iki işaretleyici arasında kalan bütün konumlar bu kipten etkilenir. Bu işaretleyicilere geleneksel olarak **sihirli çerezler** denir.

Görünüm kiplerini açıp kapamak amacıyla her iki uçbirim sınıfında da (sihirli çerezli uçbirimler ve çerezsiz uçbirimler) aynı yetenekler (**so**, **se**, **mb**, vb.) kullanılır. Sihirli çerezli uçbirimlerde bu komutlar sihirli çerezleri yazarlar. Çerezsiz uçbirimlerde ise çıktığı etkileyecek o anki kipleri değiştirirler. Basit uygulamalar bu komutları çerez yazıp yazmadıklarını bilmeksizin kullanabilirler.

Ancak, bir ekranda çalışan ve bunu güncelleyen bir uygulama, uçbirimin sihirli çerezleri kullanıp kullanmadığını ve bunların etkilerini tam olarak bilmek zorundadır. Bu bilgi **sg** yeteneğinden gelir.

sg yeteneği, varlığı uçbirimin görünüm kipleri için sihirli çerezleri kullandığını belirten sayısal değerli bir yetenektir. Değeri bir sihirli çerezin işgal ettiği karakter konumu sayısıdır. Genellikle çerez ekranda bir veya daha fazla karakterlik yer işgal eder ve bu karakter konumları boş olarak gösterilir. Bazı uçbirimlerde ise çerez sıfır genişlikte olabilir.

sg yeteneği çarpıcı görünüm kipini açan ve kapayan çerezlerin her ikisi için de geçerlidir. Bu, her iki çerez çeşidinin de ekranda aynı genişlikte olacağı varsayımına dayanır. Eğer bu gerçekleşmezse, daha dar çerezler diğer çerezlerle aynı genişliğe gelene kadar boşluklarla "genişletilir".

Bazı sihirli çerezli uçbirimlerde, her satır daima normal görünümle başlar; başka bir deyişle bir sihirli çerezin etki alanı satırın dışına çıkamaz. Fakat diğer uçbirimlerde bir sihirli çerez açıkça iptal edilmedikçe kendinden sonraki satırları da etkiler. Termcap sihirli çerezlerin bu anlamda çalışma yöntemlerini ayırmsamaya yarayacak bir yol tanımlamaz. Sonuçtan emin olmak için en iyisi her satırın başına bir çerez koymaktır.

Bazı çerezsiz uçbirimlerde, çarpıcı görünüm kipi veya diğer görünüm kipleri imleç hareket ettirilerek iptal edilebilir. Bazılarında da, imlecin hareket ettirilmesinin görünüm kipleri üzerinde bir etkisi olmaz. Bu ikinci tür uçbirimler seçimlik **ms** ("can move in standout" kısaltması olarak) yeteneği ile tanınırlar. Görünüm kipleri etkinken imlecin taşınmasına sebep olabilecek durumlara sahip tüm uygulamalar bu seçimlik yeteneğin varlığına bakmalıdır; yeteneğin yokluğu durumunda imleci hareket ettirmeden önce görünüm kiplerini normale getirmelidirler.

Bir uygulama sadece çarpıcı görünüm kipini etkinleştirmişse kipi normale çevirmek için **se** yeteneğini kullanmalıdır. Sadece diğer karakter kümesi kipini etkinleştirmişse kipi normale çevirmek için **ae** yeteneğini kullanmalıdır. Eğer diğer görünüm kiplerini etkinleştirmişse bu kipleri normale çevirmek için **me** yeteneğini kullanmalıdır.



Bilgi

so ve **mb ... mr** komutlarından biri ile bir görünüm kipi eğer diğer görünüm kiplerinden biri etkinken etkinleştirilmeye çalışılırsa bazı uçbirimlerde iki kip birleştirilirken bazılarında evvelce etkin olan kip kapatılır. İkinci tür uçbirim davranışının sebebi bu uçbirimlerin başka bir görünüm kipine geçmeksizin bir görünüm kipini etkinleştirecek ya da kapatacak komutlarının olmayışdır. Görünüm kiplerini birarada kullanacak uygulamalar bunu **sa** yeteneğinin yokluğunda yapmaya çalışmamalı, bir kipten diğerine geçerken evvelce etkin olan kipi kapatmalı ve ondan sonra istenen kipe geçmelidirler.

Bazı eski uçbirimlerde **so** ve **se** komutları yerine çarpıcı görünüm kipini açıksa kapatan kapalıysa açan komut olarak tek bir komut olabilir. Bu durum kurallara uyma adına ek bir **se** komutu çıkıtılayacak bir uygulama için risklidir. Şansa bakın ki, bu uçbirimlerin hepsi artık atıl olmuştur.

Çarpıcı görünümde olmayan metni çarpıcı görünümle değiştirerek ekranı güncelleyen uygulamalar seçimlik **xs** yeteneğinin varlığına bakmalıdır. Çerezsiz uçbirimlerde, bu yetenek yazılmış çarpıcı görünüm kipinden çıkmanın tek yolunun **ce** dizgesi ile ya da daha güçlü bir şeyle satırın bu bölümünün temizlenmesi olduğunu söyler (bkz. [Ekran Parçalarının Temizlenmesi](#) (sayfa: 28)); bu ekran konumlarına sadece yeni karakterler yazmakla orada etkin olan kipler değişmeyecektir. Sihirli çerezli uçbirimlerde ise **xs** yeteneği bir çerezi kaldırmanın tek yolunun çerezi içeren satır bölümünü temizlemek olduğunu söyler; aynı konuma farklı bir çerez yazmak işe yaramaz.

Böyle uygulamalar ayrıca uçbirimin bir Teleray 1061 olduğunu belirten seçimlik **xt** yeteneğinin varlığına da bakmalıdır. Bu uçbirimde imleci bir sihirli çerezin önüne konumlamak mümkündür, bu durumda bir çerezi kaldırmak için iki yol olur: (1) imlecin üzerinde bulunduğu satırı silmek, (2) imleci çerezin konumundan en az bir karakter önceye (muhtemelen bir önceki satıra) konumlamak ve bu uçbirimlerde sonraki **so** sihirli çerezini bulup kaldıran **se** dizgesini çıkıtılamaktır. (Ayrıca, bir satırın başlangıcında olmayan bir çerezi o satırı temizleyerek de kaldırmak mümkün olabilir.) **xt** yeteneğinin ayrıca, sekme karakterlerinin kullanımıyla da sorunları vardır, fakat bu bakımdan artık atıl olmuştur (bkz. [İmleç Hareketleri](#) (sayfa: 21)).

so

Çarpıcı görünüm kipine girmeyi sağlayan dizge değerli yetenek.

se

Çarpıcı görünüm kipinden çıkmayı sağlayan dizge değerli yetenek.

sg

Sihirli çerezli ekranın genişliğini veren sayısal yetenek. Görünüm kipini karakter karakter kaydeden uçbirimlerde bu yetenek yoktur.

ms

Varlığı, görünüm kipleri normal durumda değilken imleci hareket ettirmenin güvenli olduğu anlamına gelen seçimlik yetenek. Bu seçeneğin yokluğunda, uygulamalar imleci hareket ettirmeden önce daima görünüm kiplerini normale çevirmelidir.

xs

Varlığı, ekrandaki görünüm kiplerini sıfırlamanın tek yolunun satırı sonuna kadar temizlemek olduğu anlamına gelen seçimlik yetenek. Çerezsiz uçbirimlerde, kiplerin etkin olduğu alanı temizlemelisiniz. Sihirli çerezli uçbirimlerde ise çerezi içeren alanı temizlemelisiniz. Ayrıntılar için yukarıya bakınız.

xt

Varlığı, imlecin tam sihirli çerezin önüne konumlanamayacağını ve imleçten sonraki sihirli çerezi silecek komutun **se** olduğunu belirten seçimlik yetenek. Ayrıntılar için yukarıya bakınız.

mb

Yanarsöner kipe girmeyi sağlayan dizge değerli yetenek.

md

Çifte parlaklık kipine girmeyi sağlayan dizge değerli yetenek.

mh

Yarı parlaklık kipine girmeyi sağlayan dizge değerli yetenek.

mk

Görünmez kipe girmeyi sağlayan dizge değerli yetenek.

mp

Korumalı kipe girmeyi sağlayan dizge değerli yetenek.

mr

Negatif görüntü kipine girmeyi sağlayan dizge değerli yetenek.

me

Çarpıcı görünüm kipi ve altçizme kipi dahil tüm görünüm kiplerini kapatan dizge değerli yetenek. Bazı uçbirimlerde ayrıca, diğer karakter kümesi kipini de kapatırken bazılarında kapatmayabilir. **mb ... mr** yeteneklerinden birinin mevcudiyeti bu yeteneğin de mevcudiyetini gerektirir.

as

Diğer karakter kümesi kipini etkin kılan dizge değerli yetenek. Bu kip karakterlerin bazıları ya da bir kısmı için başka bir resim atar. Bu karakter resimlerinin ne olacağı ile ilgili bir standart yoktur.

ae

Diğer karakter kümesi kipini kapatan dizge değerli yetenek.

sa

Görünüm kiplerinin keyfi bir birleşimini açmaya yarayan dizge değerli yetenek. Herbiri farklı bir görünüm kipini temsil eden 9 parametre kabul eder. Değeri 1 olan görünüm kipi etkin, 0 olan kapalıdır. Uçbirimlerin çoğu **sa** yeteneğini desteklemez.

Dokuz parametre sırasıyla, *çarpıcı görünüm*, *altçizme*, *negatiflik*, *yanarsönerlik*, *yarı parlaklık*, *çifte parlaklık*, *boş*, *korumalı*, *diğer karakter kümesi* kiplerini temsil eder.

4.11. Altçizme

Çoğu uçbirimde altçizme de çarpıcı görünüm kipi gibi bir görünüm kipi çeşididir. Bu sebeple sihirli çerezler kullanılarak ya da çıktılanan her karakterin anlık durumunu etkileyen bir seçenek olarak gerçekleştirilebilir. Ayrıntılı bilgi için *Çarpıcı ve Sıradan Görünüm Kipleri* (sayfa: 33) bölümüne bakınız.

ug yeteneği, uçbirimin altçizme için sihirli çerezler kullandığını belirten sayısal bir yetenektir ve değeri altçizme çerezinin işgal ettiği karakter konumlarının sayısıdır. **sg**'nin çarpıcı görünüm kipi olarak kullanılması gibi bu yetenek de altçizme kipi için kullanılır. Çok basit uygulamalar dışında, **ug**'nin değerini dikkate almaksızın altçizme kipini doğru olarak kullanmak imkansızdır.

us

Altçizme kipini etkin kılan ya da altçizme kipini başlatan bir sihirli çerez çıkıtlayan dizge değerli yetenek.

ue

Altçizme kipini etkisiz kılan ya da altçizme kipini kapatpan bir sihirli çerez çıkıtlayan dizge değerli yetenek.

ug

Altçizme kipiyle ilgili bir değişikliği belirten sihirli çerezin genişliği; eğer uçbirim bunun için sihirli çerezleri kullanmıyorsa bu yetenek yoktur.

ms

Varlığı, görünüm kipleri normal durumda değilken imleci hareket ettirmenin güvenli olduğu anlamına gelen seçimlik yetenek. Bu seçeneğin yokluğunda, uygulamalar imleci hareket ettirmeden önce daima altçizme kipini kapatmalıdır.

Altçizme yapmak için iki eski yöntem daha vardır: (1) tek bir karakteri altçizgili yapan bir komut vardır, (2) altçizgi karakterinin () çıkıtlanması ve üste yazmama yöntemiyle bir karakterin altı çizilebilir. Bunları destekleyen gözden ırak kalmış uçbirimleri bulup çıkarmak hususunda yazarı çaba harcamadıkça yeni uygulamalar bu yetenekleri yakalamakla uğraşmazlar. Yine de uçbirim tanımları uygun olduğunda bu yetenekleri sağlamalıdır.

uc

İmlecin altındaki karakterin altını çizdikten sonra imleci sağa taşıyan dizge değerli yetenek.

ul

Varlığı, uçbirimin altçizme işlemini karaktere altçizgi karakterini () ekleyerek yapabildiğini belirten seçimlik yetenek; genelde bazı uçbirimler altçizme kipini desteklemiyor olsalar bile bunu yapabilirler. Bu yeteneğin olumsuz yanı yeni bir metin çıkıtlanırken eskisinin üzerine yazarak bozuk çıktı üretmesidir. Bu sebepten altçizgi karakterlerinin kendileri çıkıtlanacaksa eski metni altçizgili yapmasın diye özel olarak ele alınmalıdır.

4.12. İmleç Görünürlüğü

Bazı uçbirimler imleci görünmez yapma veya genişletme yeteneğine de sahiptir. Kullanıcının ekranda ilgilendiği bir konumu daha belirgin yapmak amacıyla kullanmayı tasarlayan uygulamalarca imleç genişletmesi sıkça yapılır; örneğin Emacs girdi üzerinde imleci genişletir. Böyle uygulamalar çıkışta imleci daima eski haline getirmelidir.

vs

İmleci genişleten dizge değerli yetenek.

vi

İmleci görünmez yapan dizge değerli yetenek.

ve

İmleci normale döndüren dizge değerli yetenek.

vs veya **vi** yeteneği tanımlıysa **ve** yeteneği de tanımlanmış olmalıdır.

4.13. Çan

Burada uçbirimin kullanıcının dikkatini çekmek için kullanılan yeteneklerinden bahsedeceğiz.

b1

Uçbirimin duyulabilen bir ses çıkarmasına sebep olan dizge değerli yetenek. Bu yeteneğin yokluğunda uçbirimin bir ses çıkarabilmesinin bir yolu yoktur.

vb

Dikkat çekmek için ekranın yanıp sönmesine (görünür çan) sebep olan dizge değerli yetenek. Bu yeteneğin yokluğunda uçbirimin böyle bir şey yapabilmesinin bir yolu yoktur.

4.14. Tuştakımı ve İşlev Tuşları

Çoğu uçbirim belirli karakter dizilerini bilgisayara gönderen ok ve işlev tuşlarına sahiptir. Doğru dizilimler uçbirime bağımlı olduğundan termcap, yetenekleri kullanılan dizilimlere göre tanımlar. Termcap'in çoğu dizge değerli yeteneğinin tersine bu yetenekler uçbirime dizge gönderen komutlar değildirler.

Tuştakımının ok tuşlarını kullanacak uygulamalar **ks** yeteneğinin varlığına bakmalı ve tuştakımı ok tuşlarını etkin kılmak için uçbirime bunu göndermelidir. Böyle uygulamalar ayrıca, çıkışta **ke** dizgesini göndermelidir.

ks

Tuştakımı ok tuşlarını etkin kılan dizge değerli yetenek. Uçbirimde bu yetenek yoksa ancak bu bölümdeki diğer yetenekler mevcutsa uygulamalar tuştakımı ok tuşlarının daima etkin olduğunu varsayabilir.

ke

Tuştakımı tuşlarını normale döndüren dizge değerli yetenek. Bu yetenek sadece **ks** varsa vardır.

k1

Girdi karakterlerinin sola ok tuşuna basarak gönderilmesini sağlayan dizge değerli yetenek. Bu yeteneğin yokluğunda uçbirimin bilgisayara sola ok tuşu ile bir şey aktarabileceğini umamazsınız.

kr

Girdi karakterlerinin sağa ok tuşuna basarak gönderilmesini sağlayan dizge değerli yetenek.

ku

Girdi karakterlerinin yukarı ok tuşuna basarak gönderilmesini sağlayan dizge değerli yetenek.

kd

Girdi karakterlerinin aşağı ok tuşuna basarak gönderilmesini sağlayan dizge değerli yetenek.

kh

Girdi karakterlerinin ev konumu tuşuna basarak gönderilmesini sağlayan dizge değerli yetenek.

K1 ... K5

Klavyede üçe üçlük takım varsa, girdi karakterlerinin üçe üçlük takımının içindeki diğer beş tuşa basarak gönderilmesini sağlayan dizge değerli yetenekler. Bu tuşlardan birinin ev konumu tuşu olabileceğini ve dolayısıyla bu yeteneklerden birinin **kh** yeteneği ile aynı olacağına dikkat edin.

k0

Girdi karakterlerinin 10. işlev tuşuna (0 etiketli tuş varsa 0. tuşa) basarak gönderilmesini sağlayan dizge değerli yetenek.

k1 ... k9

Girdi karakterlerinin 1'den 9'a kadar ki işlev tuşlarına basarak gönderilmesini sağlayan dizge değerli yetenekler.

kn

10'dan fazla olduğunda işlev tuşlarının sayısını belirten sayısal yetenek.

l0 ...l9

Klavyenin **k0** ... **k9** yetenekleriyle tanımlanan tuşlarının etiketlerindeki dizgeler. Bu dizgeler zaten **f0** veya **f10** ve **f1** ... **f9** iseler bu yetenekler tanımlanmamış olmalıdır.

kH

Girdi karakterlerinin aşağı ev konumu tuşuna basarak gönderilmesini sağlayan dizge değerli yetenek.

kb

"Gerisilme" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

ka

"Tüm sekmelere temizle" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kt

"Bu sütundaki sekme durağını kaldır" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kC

Ekranı temizle" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kD

"Karakteri sil" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kL

Satırı sil" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kM

"Ekleme kipinden çık" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kE

"Satır sonuna kadar temizle" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kS

"Ekran sonuna kadar temizle" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kI

"Karakter ekle" ya da "ekleme kipine gir" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kA

"Satır ekle" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kN

"Sonraki sayfa" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kP

"Önceki sayfa" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kF

"İleri kaydır" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kR

"Geri kaydır" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

kT

"Bu sütuna sekme durağı koy" tuşu varsa, girdi karakterlerinin bu tuşa basarak gönderilmesini sağlayan dizge değerli yetenek.

ko

Uçbirimin sahip olduğu diğer işlev tuşlarını listeleyen dizge değerli yetenek. Bu, **kH** ... **kT** yeteneklerindeki bilgiyi belirtmenin artık kullanılmayan bir yoludur. Dizge, virgüllerle ayrılmış olarak iki karakterlik termcap yetenek isimlerini listeler. Listelenmiş her yetenek ismi, uçbirimin bu yeteneğin değeri olan dizgeyi gönderme yeteneğinin olduğunu belirtir. Örneğin, **:ko=c1, ll, sf, sr**: değeri, uçbirimin "ekranı temizle", "aşağı ev", "ileri kaydır" ve "geri kaydır" isimli dört işlev tuşunun olduğu anlamına gelir.

4.15. Meta Tuşu

Meta tuşu basılan her karaktere 0200 bitini ekleyerek değiştiren bir tuştur. Bu bit sadece ve sadece karakter tuşlanırken Meta tuşu basılı tutulduğunda etkilidir. Meta tuşu ile birlikte tuşlanan karakterlere Meta karakterler denir. Emacs Meta karakterleri metin düzenleme komutları olarak kullanır.

km

Varlığı, uçbirimin bir Meta tuşuna sahip olduğunu belirten seçimlik yetenek.

mm

Meta tuşunu işlevselleştiren dizge değerli yetenek.

mo

Meta tuşunu işlevsizleştiren dizge değerli yetenek.

Eğer uçbirim **km** yeteneğine sahipse ama **mm** ve **mo** yeteneklerine sahip değilse, bu, Meta tuşunun daima işlevsel olduğu anlamına gelir. **mm** ve **mo** yeteneklerine sahipse Meta tuşu açılıp kapatılabilir. **mm** göndererek işlevselleştirilir ve **mo** göndererek işlevsizleştirilir. Meta tuşunu neden işlevsizleştirmek isterler ki.

4.16. İklendirme

ti

İmleci ekranda rasgele hareket ettiren uygulamalar için uygun ya da gerekli olan her ne özel kip varsa uçbirime koyan dizge değerli yetenek. Termcap'le tam ekran çalışan uygulamalar her başlatılışlarında bu dizgeyi çıktılmalıdır.

te

ti yeteneği ile yapılan her ne varsa geri alan dizge değerli yetenek. Başlatıldıklarında **ti** çıktılardan uygulamalar çıkarken de bu yeteneği çıktılmalıdır.

is

Her oturum açmada uçbirimi ilklendirmeye yarayan dizge değerli yetenek.

if

Her kullanım oturumu için uçbirimi ilklendirecek komut dizgelerini içeren dosyanın ismini içeren dizge değerli yetenek. Normalde **is** ve **if** birlikte kullanılmaz.

i1

i3

Her oturum açmada uçbirimi ilklendirmeye yarayan dizge değerli iki yetenek daha. **i1** tanımlıysa **is**'den önce, **i3** tanımlıysa **is**'den sonra çıktılır.

Üç farklı ilklendirme dizgesinin oluş sebebi uçbirimle ilgili bir grup tuşu biraz farklı ilklendirebilmeyi kolaylaştırmaktır. İki veya üç dizge temel türde tanımlanır; sonradan diğer türler bir veya iki dizgeyi değiştirebilir.

rs

Tuhaf bir kipteki uçbirimi sınırlamaya yarayan dizge değerli yetenek. Normalde bu yetenek **is** (veya aynı etkiye sahip başka yetenekler) ve diğer birtakım yetenekleri de içerir. **is** içermeyen bir **rs** yeteneği, tuhaf bir kipteki uçbirimi sınırlamak için hiç kimsenin kullanmak istemeyeceği can sıkıcı ve yavaş bir komut olurdu.

it

Uçbirim enerjilendiğinde donanımsal sekme durakları arasındaki ilk boşluk miktarını veren sayısal yetenek. Uçbirimi ilklendiren uygulamalar sekme duraklarının ayarlanmasının gerekli olup olmadığına bu yeteneğe bakarak karar verirler. İlk genişlik 8 ise doğru ve iyidir, değilse duraklar ayarlanmalıdır. Ayarlanamıyorsa, sekmeleri boşluklara çevirmesi için çekirdeğe başvurulur.

ct

Tüm sekme duraklarını yokeden dizge değerli yetenek.

st

Tüm satırlar için o anki imleç sütununu sekme durağı yapan dizge değerli yetenek.

NF

Varlığı, uçbirimin XON/XOFF akış denetimini desteklemediğini belirten seçimlik yetenek. Bu durumda uygulamalar uçbirime XON (**C-q**) veya XOFF (**C-s**) karakterlerini göndermemelidirler.

4.17. Geciktirme Yetenekleri

Komut dizgesi olarak kullanılan tüm yeteneklerin geciktirme belirtilmelerine riayet etmenin doğru yolunu açıklayan sadece iki uçbirim yeteneği vardır. Biri **pc**'dir ve termcap kullanan tüm uygulamalarca riayet edilmelidir.

pb

Geciktirmenin gerektiği en düşük aktarım hızının (baud cinsinden) belirtildiği sayısal yetenek. Uygulamalar buna bakarak düşük hızlarda geciktirme yapmaktan kaçınabilirler.

pc

Geciktirme için kullanılan dizge değerli yetenek. Bu dizgenin ilk karakteri adım karakteri olarak boş (null) karakter yerine kullanılır. Bu yeteneğin olmadığı durumda adım karakteri olarak boş karakter kullanılır. Termcap kullanan her uygulama bu yeteneğe bakmalı ve onu **tputs** işlevinin kullandığı **PC** değişkenini ayarlamakta kullanmalıdır. Bkz. [Geciktirme](#) (sayfa: 8).

Sıradan ardışık çıktıda kullanılan imleç hareket komutlarına çekirdeğin verdiği geciktirme miktarının belirtildiği bazı termcap yetenekleri vardır.

dC

Satırbaşı karakteri (CR) için gerekli geciktirmenin ms cinsinden değerini içeren sayısal yetenek.

dN

Satırsonu karakteri (LF) için gerekli geciktirmenin ms cinsinden değerini içeren sayısal yetenek.

dB

Gerisilme karakteri için gerekli geciktirmenin ms cinsinden değerini içeren sayısal yetenek.

dF

Sayfa ileri karakteri için gerekli geciktirmenin ms cinsinden değerini içeren sayısal yetenek.

dT

Sekme karakteri için gerekli geciktirmenin ms cinsinden değerini içeren sayısal yetenek.

Bazı sistemlerde çekirdek yukarıdaki yetenekleri kullanırken bazılarında geciktirme belirten dizge değerli **cr**, **sf**, **le**, **ff** ve **ta** yeteneklerini kullanır. Böyle geciktirme gerektiren uçbirim tanımları **dC** ... **dT** yeteneklerini içermeli ve ayrıca ilgili dizge değerli yeteneklerde uygun geciktirme miktarını belirtmelidir. Sıradan ardışık çıktı için geciktirme gerektirmeyen günümüz uçbirimlerinde belki de böyle şeyler yapmaya ihtiyaç duymayacaksınız.

4.18. Durum Satırı

Durum satırı uçbirim üzerinde normal çıktı için değil, sadece özel iletiler için kullanılan bir satırdır. Tasarlandığı haliyle uygulamanın ne yapmakta olduğunu kullanıcıya açıklamak için sürekli güncellenen bir satırdır ve bu da bu satıra neden bu ismin verildiğini açıklar. Fakat aslında herşey için kullanılabilirdi. Durum satırının uçbirim karakteristiklerinin farklılaştırılması sıradan çıktıyı etkilemez. Durum satırının içeriği sadece bu bölümde açıklanan özel durum satırı yetenekleriyle değiştirilir.

hs

Varlığı, uçbirimin bir durum satırı olduğunu belirten seçimlik yetenek. Bir durum satırı içeren bir uçbirim **ts** ve **fs** yeteneklerini de sağlamalıdır.

ts

Uçbirim imlecini durum satırına taşıyan dizge değerli yetenek. Bu yetenekler **fs** yeteneğinin amacına uygun olarak eski imleç konumunu özellikle kaydetmelidir.

fs

İmleci durum satırından alıp eski konumuna taşıyan dizge değerli yetenek.

es

Varlığı, durum satırına yazılırken diğer ekran komutlarının çalışacağını belirten seçimlik yetenek. Başka bir deyişle, durum satırı kısmen temizlenirken, aynı zamanda çarpıcı görünüm kipine girmek ve bırakmak için **ch** kullanmak üzere imleç hareket ettirilebilir, karakterler silinebilir ve eklenebilir (şüphesiz bir **ch** yeteneği mevcutsa).

ds

Durum satırının gösterimini iptal eden dizge değerli yetenek. Bunun yokluğunda durum satırını ekrandan kaldırmanın bir yolu yoktur.

ws

Durum satırının genişliğini içeren sayısal yetenek. Durum satırı olan bir uçbirimde bu yetenek yoksa durum satırı diğer satırlarla aynı genişlikte demektir.

ws değerinin bazan 8 gibi küçücük bir değer olabileceğini aklınızdan çıkarmayın.

4.19. Yarım Satırlık Hareket

Bazı uçbirimler alt ve üst indisleri çıkıtlayabilmek için imleci düşeyde yarım satır hareket ettirme yeteneğine sahiptirler. Çoğunlukla yazıcı benzeri uçbirimlerin böyle yetenekleri vardır.

hu

İmleci yarım satır yukarı taşıyan dizge değerli yetenek. Eğer uçbirim ekransa sorumluluğunuz üst satıra taşmamaktır; yine de bunu destekleyen çoğu uçbirim, yazıcı benzeri basılı çıktı veren bir aygıttır ve endişelenmeniz gerekmez.

hd

İmleci yarım satır aşağı taşıyan dizge değerli yetenek. Eğer uçbirim ekransa sorumluluğunuz alt satıra taşmamaktır.

4.20. Uçbirimlerle İlintili Yazıcıların Denetimi

Bazı uçbirimlere yazıcı benzeri basılı çıktı veren aygıtlar bağlıdır. Bunlar ekran içeriğini yazıcıya kopyalayabilecekleri gibi çıktıyı yazıcıya da yönlendirebilirler. Termcap bir yazılıma ekran üzerinde çıktı göründükçe bunların da yönlendirileceğini belirtecek bir yola sahip değildir; bazı uçbirimler bunu yapabilir, bazıları yapmaz.

ps

Ekran içeriğinin yazdırılmasını sağlayan dizge değerli yetenek. Yokluğunda ekran içeriği yazıcıya çıktılanamaz.

po

Çıktıyı yazıcıya yönlendiren dizge değerli yetenek.

pf

Çıktının yazıcıya yönlendirilmesini sonlandıran dizge değerli yetenek. Uçbirim tanımında **po** mevcutsa bu yetenek de olmak zorundadır.

pO

Çıktının sonraki *n* karakterini ne olduklarıyla ilgilenmeksizin yazıcıya yönlendiren dizge değerli yetenek. Yönlendirme *n* karakter gönderildikten sonra sona erer. *n* karakter gönderilinceye kadar **pf** yeteneği olsa bile hiçbir şey yönlendirmeyi sonlandıramaz. *n* hiçbir zaman 255'ten büyük olamaz.

Bu yeteneğin kullanım amaçlarından biri de metin olmayan bayt dizilerinin de (biteşlemler gibi) yazıcıya gönderilebilmesidir.

Yazıcılarında içinde bulunduğu çoğu uçbirim **ps**, **po** ve **pO** yeteneklerinin hepsini birden desteklemez; çıktısını her çeşit yazıcıya gönderebilen bir uygulamanın bu üç yeteneğin de varlığına bakması ve içlerinden en uygun olanı seçip kendi amacına uygun olarak kullanması gerekir.

5. Yetenek İsimlerinin Listesi

Burada yetenek isimleri kısa açıklamalarıyla birlikte alfabetik sırada verilmiştir. Tanımları ile ilgili konu başlıklarına [Yetenekler Dizini](#) (sayfa: 54) üzerinden ulaşabilirsiniz.

5.1. Seçimlik Yetenekler

am

Son sütuna yazılan karakterin imlecin sonraki satırın başına taşınmasına sebep olacağını belirten seçimlik yetenek.

bs

Atıl: Sola hareket için de kullanılabilen ASCII gerisilme karakteri.

bw

Sol girintide kullanıldığında **le** komutunun imleci önceki satırın son sütununa taşıyacağını belirten seçimlik yetenek.

da

Satırların yukarı kaydırılması sırasında ilk satır aşıldığında, eğer ardından aşağı kaydırma yapılırsa ekranın geri gelebileceğini belirten seçimlik yetenek.

db

Satırların aşağı kaydırılması sırasında son satır aşıldığında, eğer ardından yukarı kaydırma yapılırsa ekranın geri gelebileceğini belirten seçimlik yetenek.

eo

Üste yazmama özelliği etkin olsa bile çıktılanan boşluk karakteri aynı konumdaki karakteri siler.

es

Durum satırına yazma sırasında diğer ekran komutlarının çalışabildiğini gösterir.

gn

Uçbirimin gerçek değil soysal bir tür olduğunu belirtir.

hc

Uçbirim tanımının bir yazıcıya ait olduğunu belirtir..

hs

Uçbirimin bir durum satırına sahip olduğunu belirtir.

hz

Uçbirimin ~ karakterini çıktı olarak kabul edemediğini belirtir.

in

Boşluk karakteri konumlarının boş konum olarak ele alınmadığını belirten yetenek.

km

Uçbirimin bir Meta tuşu vardır.

LP

Son satırın son sütununa yazmak ekranı kaydırmaz.

5i

Yazıcıya gönderilen veri ekrana yansıtılmaz.

mi

Ekleme kipinde imleç hareketi güvenlidir.

ms

Çarpıcı görünüm kipinde imleç hareketi güvenlidir.

nc

Atıl: bu uçbirimde ASCII satırbaşı (CR) kullanılmaz.

NF

XON/XOFF akış denetimi kullanılmaz.

NP

Geciktirme karakteri gerekmez.

NR

ti yeteneği **te** yeteneğinin tersi değildir.

ns

Uçbirimin sıradan çıktığı normalde kaydırmayacağını belirten seçimlik yetenek.

nx

Geciktirme çalışmaz. XON/XOFF akış denetimi gerekir.

os

Uçbirimin üste yazmama özelliği olup olmadığını belirleyen seçimlik yetenek.

ul

Altçizgili karakterler üzerine yazılarak altı çizili yapılır.

xb

<ESC> ve <Control>-<C> karakterlerini gönderemeyen Superbee uçbirim türünü tanımlayan seçimlik yetenek.

xn

İmleci sarmalamanın tuhaf bir yolu; 80 sütundan sonra satırsonu karakteri yoksayılr.

xo

XON/XOFF akış denetimi kullanılır.

xs

Bir satırı temizlemek için tek yolun bu satırdaki görünüm kipi konumlarının (veya sihirli çerezlerin) temizlenmesi olduğunu belirten yetenek.

xt

Teleray 1061; muhtelif tuhaf karakteristikler.

5.2. Sayısal Yetenekler

co

Ekran genişliği.

dB

Atıl: Gerisilme karakteri için gereken gecikme (ms cinsinden).

dC

Atıl: Satırbaşı karakteri (CR) için gereken gecikme (ms cinsinden)

dF

Atıl: Sayfa ileri karakteri için gereken gecikme (ms cinsinden)

dN

Atıl: Satırsonu karakteri (LF) için gereken gecikme (ms cinsinden)

dT

Atıl: Sekme karakteri için gereken gecikme (ms cinsinden).

it

Donanımsal sekme durakları arasındaki boşluğun sütun sayısı cinsinden ilk değeri.

kn

İşlev tuşunun numarası.

li

Ekranın yüksekliği.

lm

Ekran belleğinin satır sayısı.

pb

Geciktirme için aslen gerekli en düşük aktarım hızı (baud rate).

sg

Sihirli çarpıcılık çerezinin genişliği. Sihirli çerezler kullanılmıyorsa yoktur.

ug

Sihirli altçizme çerezinin genişliği. Altçizme için sihirli çerezler kullanılmıyorsa yoktur.

ws

Durum satırının genişliği.

5.3. Dizgesel Yetenekler

ae

Diğer karakter kümesi kipini kapatan komut.

al

İmleç konumunun öncesine boş bir satır yerleştiren komut.

AL

İmleç konumunun öncesine *n* boş satır yerleştiren komut.

as

Diğer karakter kümesi kipini etkinleştiren komut.

bc

le komutunun çok eski ve artık kullanılmayan ismi,

bl

Zil sesi komutu.

bt

İmleci önceki donanımsal sekme durağının bulunduğu sütunun soluna taşıyan komut.

CC

Uçbirimin komut karakterini değiştiren komut.

cd

İmlecin bulunduğu satır dahil sonraki satırları temizleyen komut.

ce

İmleç konumundan satırın sonuna kadar silen komut.

ch

İmleci aynı satırın *c*'inci sütununa konumlayan komut.

cl

Ekranı tamamen temizleyip imleci sol üst köşeye konumlayan komut.

cm

İmleci *l*'inci satırın *c*'inci sütununa konumlayan komut.

CM

Ekran belleğine göreli olarak, imleci *l*'inci satırın *c*'inci sütununa konumlayan komut.

cr

İmleci üzerinde bulunduğu satırın başına taşıyan komut.

cs

Kayıdırma bölgesinin belirtilebilmesini sağlayan komut.

cS

Kayıdırma bölgesinin belirtilebilmesini parametreleri farklı bir biçimde kullanarak sağlayan komut.

ct

Tüm sekme duraklarını kaldıran komut.

cv

İmleci aynı sütunda *l*'inci satıra taşıyan komut.

dc

İmlecin bulunduğu konumdaki karakteri silen komut.

DC

İmlecin bulunduğu konumdan başlayarak *n* karakteri silen komut.

dl

İmlecin bulunduğu satırı silen komut.

DL

İmlecin bulunduğu satırdan başlayarak n satırı silen komut.

dm

Silme kipine giriş komutu.

do

İmleci bir satır aşağı taşıyan komut.

DO

İmleci n satır aşağı taşıyan komut.

ds

Durum satırının gösterimini iptal eden komut.

ec

İmlecin üzerinde bulunduğu karakterden başlayarak n karakteri temizleyen komut.

ed

Silme kipinden çıkış komutu.

ei

Ekleme kipini bırakma komutu.

ff

Yazıcı için, sonraki sayfaya geçiş komutu.

fs

İmlecin son satırın başlangıcında olduğu kabulüyle, ekran içeriğini bir satır yukarı kaydıran komut.

hd

İmleci yarım satır aşağı taşıyan dizge.

ho

İmleci sol üst köşeye konumlayan komut.

hu

İmleci yarım satır yukarı taşıyan dizge.

i1

Her yeni oturum açılışında uçbirimi ilklendiren komut.

i3

Yeni oturum açılışı için ilklendirme komutu.

ic

İmleç konumuna bir karakter ekleyen komut.

IC

İmleç konumuna n karakter ekleyen komut.

if

Uçbirimi ilklendirecek komutların bulunduğu dosyanın isminin belirtildiği komut.

im

Ekleme kipine giriş komutu.

ip

Ekleme kipinde eklenen karakterden sonrasını çıktılıyan komut.

is

Her yeni oturum açılışında uçbirimi ilklendiren komut.

k0

10. (varsa 0.) işlev tuşu ile girdi gönderme komutu.

k1 ... k9

1.'den 9.'ya kadar işlev tuşları ile karakter gönderme komutları.

K1 ... K5

Üçe üçlük tuştakımının içindeki diğer beş tuş için karakter gönderme komutları.

ka

"Tüm sekmeleri temizle" tuşu ile girdi gönderme komutu.

kA

"Ekleme kipi" tuşu ile girdi gönderme komutu.

kb

"Gerisilme" tuşu ile girdi gönderme komutu.

kC

"Ekranı temizle" tuşu ile girdi gönderme komutu.

kd

Aşağı ok tuşu ile girdi gönderme komutu.

kD

"Silme karakteri" tuşu ile girdi gönderme komutu.

ke

İşlev tuşlarını yerel olarak çalıştır yapma komutu.

kE

"Satır sonuna kadar sil" tuşu ile girdi gönderme komutu.

kF

"İleri kaydır" tuşu ile girdi gönderme komutu.

kh

"Eve git" tuşu ile girdi gönderme komutu.

kH

"Aşağı ev" tuşu ile girdi gönderme komutu.

kI

"Karakter ekle" veya "Ekleme kipine gir" tuşu ile girdi gönderme komutu.

kl

Sola ok tuşu ile girdi gönderme komutu.

kL

"Satırı sil" tuşu ile girdi gönderme komutu.

kM

"Ekleme kipinden çık" tuşu ile girdi gönderme komutu.

kN

"Sonraki sayfa" tuşu ile girdi gönderme komutu.

kO

İyice Atıl: uçbirimin isimli işlev tuşlarını listeleme komutu.

kP

"Önceki sayfa" tuşu ile girdi gönderme komutu.

kR

Sağa ok tuşu ile girdi gönderme komutu.

kR

"Geri kaydır" tuşu ile girdi gönderme komutu.

kS

İşlev tuşlarını yayın tuşları yapan komut.

kS

"Ekran sonuna kadar temizle" tuşu ile girdi gönderme komutu.

kt

"Bu sütunun sekme durağını kaldır" tuşu ile girdi gönderme komutu.

kT

"Bu sütuna sekme durağı koy" tuşu ile girdi gönderme komutu.

ku

Yukarı ok tuşu ile girdi gönderme komutu.

lO

10. (varsa 0.) işlev tuşunu etiketleyen komut.

l1 ... l9

1'den 9'a kadar işlev tuşlarını etiketleyen komut.

le

İmleci bir sütun sola taşıyan komut.

LE

İmleci *n* sütun sola taşıyan komut.

ll

İmleci sol alt köşeye konumlayan komut.

mb

Yanardöner kipe giriş komutu.

md

Çifte parlak kipe giriş komutu.

me

Tüm görünüm kiplerini kapatan komut.

mh

Yarı parlak kipe giriş komutu.

mk

Görünmezlik kipine giriş komutu.

mm

Meta tuşunu işlevselleştirme komutu.

mo

Meta tuşunu işlevsizleştirme komutu.

mp

Korumalı kipe giriş komutu.

mr

Negatif görüntü kipine giriş komutu.

nd

İmleci bir sütun sağa taşıyan komut.

nl

Atıl: **do** ve **sf** yerine artık kullanılmayan yeteneğin ismi.

nw

Bazan satırın kalanını da silerek imleci sonraki satırın başına taşıyan komut.

pc

Geciktirme için kullanılacak karakteri belirten komut.

pf

Yazıcıyı çıktı yönlendirmeyi sonlandıran komut.

po

Çıktıyı yazıcıya yönlendiren komut.

pO

Sonraki *n* karakteri yazıcıya yönlendiren komut.

ps

Ekranı bağlı yazıcıya basan komut.

rc

İmleci kayıtlı son konuma taşıyan komut.

RI

İmleci *n* karakter sağa taşıyan komut.

rp

c karakterini *n* kere basan komut.

rs

Uçbirimi tuhaf bir kipteyken sıfırlama komutu.

sa

Görünürlük kiplerini açıp kapayan komut.

sc

	O anki imleç konumunu kaydeden komut.
se	Çarpıcı görünüm kipini bırakma komutu.
sf	Ekranı bir satır yukarı kaydıran komut.
SF	Ekranı n satır yukarı kaydıran komut.
so	Çarpıcı görünüm kipine giriş komutu.
sr	Ekranı bir satır aşağı kaydıran komut.
SR	Ekranı n satır aşağı kaydıran komut.
st	İmleç konumunu tüm satırlar için sekme durağı yapan komut.
ta	İmleci sonraki donanımsal sekme durağına taşıyan komut.
te	Uçbirimi sıralı çıktıya ayarlayan komut.
ti	Uçbirimi rasgele imleç hareketi için ilklendiren komut.
ts	Uçbirim imlecini durum satırına taşıyan komut.
uc	Karakterin altını çizdikten sonra imleci sağa taşıyan komut.
ue	Altçizme kipinden çıkış komutu.
up	İmleci bir satır yukarı taşıyan dizge.
UP	İmleci n satır yukarı taşıyan dizge.
us	Altçizme kipine giriş komutu.
vb	Ekranı yanardöner yapan komut.
ve	İmleci normale döndüren komut.
vi	

İmleci görünmez yapan komut.

vs

İmleci genişleten komut.

wi

Uçbirimin çıktılama için kullanacağı ekran penceresini ayarlamakta kullanılan komut.

Değişkenler ve İşlevler Dizini

B

BC 15

O

ospeed 10

P

PC 10

T

tgetent işlevi 5

tgetflag 6

tgetnum 6

tgetstr 6

tgoto 15

tparam 14

tputs 10, 10

U

UP 15

Yetenekler Dizini

Semboller

5i 44

A

ae 35

al 28

AL 29

am 25

as 35

B

bc 24

bl 37

bs 24

bt 24

bw 23

C

CC 20

ce 28

ch 23

cl 28

cm 22

CM 23

co 21

cr 22

cs 26

cS 26

ct 40

cv 23

D

da 27

db 27

dB 41

dc 32

DC 32

dC 41

dF 41

dl 29

DL 29

dm 32

dN 41

DO 23

ds 42

dT 41

E

ec 28

ed 32

ei 30

eo 20

es 41

F

ff 23

fs 41

G

gn 20

H

hc 20

hd 42

ho 22

hs 41

hu 42

hz 20

I

IC 31

i

i1 40

i3 40

ic 31

im 30

ip 31

is 40

it 40

K

k0 37

K1...K5 37

k1...k9 38

ka 37, 38

kA 39

kb 38

kC 38

kd 37

kD 38

ke 37

kE 38

kF 39

kh 37

kH 38

kl.....	37, 38
kL.....	38
kM.....	38
km.....	39
kN.....	39
ko.....	39
kP.....	39
kr.....	37
kR.....	39
kS.....	38
kt.....	38
kT.....	39
ku.....	37

L

l0...l9.....	38
le.....	22
LE.....	23
li.....	21
ll.....	22
lm.....	27
LP.....	25

M

mb.....	35
md.....	35
me.....	35
mh.....	35
mi.....	31
mk.....	35
mm.....	39
mo.....	39
mp.....	35
mr.....	35
ms.....	34, 36

N

nc.....	24
nd.....	22
NF.....	40
nl.....	24
NP.....	44
NR.....	44
ns.....	27
nw.....	23
nx.....	44

O

os.....	20
---------	----

P

pb.....	40
pc.....	41
pf.....	42
po.....	42
pO.....	42
ps.....	42

R

rc.....	23
Rl.....	23
rp.....	20
rs.....	40

S

sa.....	35
sc.....	23
se.....	34
sf.....	25
SF.....	26
sg.....	34
so.....	34
sr.....	26
SR.....	26
st.....	40

T

ta.....	24
te.....	40
ti.....	39
ts.....	41

U

uc.....	36
ue.....	36
ug.....	36
ul.....	36
up.....	22
us.....	36

V

vb.....	37
ve.....	36
vs.....	36

W

wi.....	27
ws.....	42

X

xb.....	21
xn.....	25
xo.....	44
xs.....	35
xt.....	24, 35

Kavramlar Dizini**A**

altçizme 35

Ç

çan 37

çıkıtıyı yinele 20

D

durum satırı 41

E

ekleme kipi 30

ekran boyutları 21

ekran boyutu 17

ekran genişliği 21

ekranın temizlenmesi 28

ev konumu 22

G

geciktirme 8, 40

görünüm kipleri 33

görünür çan 37

görünürlük 36

göze çarpma 33

H

hat hızı 10

I

ilkendirme 39

imleç hareketleri 21

K

karakter ekleme 30

karakter silme 30

kaydırma 25

komut karakteri 20

M

meta tuşu 39

miras alma 18

P

parametrelerin kodlanması 11

pencere 27

S

sarmalama 17, 25

satır ekleme 28

satır silme 28

sekme durakları 39

sıfırlama 39

sihirli çerez 33

silme 28

silme kipi 30

soysal uçbirim türü 20

Superbee 21

T

tanım biçimi 15

termcap 4

U

uçbirim seçenekleri (çekirdek) 8

uçbirim türlerinin isimleri 17

Ü

üste yazmama 20

Y

yazıcı 42

Notlar

- a) Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler buldukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.
- b) Konsol görüntüsünü temsil eden sarı zeminli alanlarda metin genişliğine sığmayan satırların sığmayan kısmı `␣` karakteri kullanılarak bir alt satıra indirilmiştir. Sarı zeminli alanlarda `␣` karakteri ile başlayan satırlar bir önceki satırın devamı olarak ele alınmalıdır.

(B1) <http://www.gnu.org/software/termutils/manual/termcap-1.3/termcap.html>

Bu dosya (termcap.pdf), belgenin XML biçiminin \TeX Live ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

27 Şubat 2007