

Sistem Programlama

Ders 11

Dr. Öğr. Üyesi Mehmet Dinçer Erbaş
Bolu Abant İzzet Baysal Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

İşlem Çevresi

- Çevre değişkenleri
 - Daha önce belirttiğimiz üzere çevre değişkenleri name=value şeklinde tanımlanır.
 - Kernel bu bilgi metinlerini asla kullanmaz.
 - Anlamları uygulamalar tarafından belirlenir.
 - Örneğin kabuk, çevre değişkenlerini kullanabilir.
 - HOME, USER vb.
 - ISO C çevre değişkenlerine erişebilmemiz için bir fonksiyon tanımlamıştır.
 - Ancak değişkenlerin içeriği sistem tarafından belirlenir.

```
#include <stdlib.h>
```

```
char *getenv(const char *name);
```

Dönüş: OK ise ismi verilen değişkenin değeri, bulunmazsa NULL.

Çevre değişkenleri

- Çevre değişkenlerinde değişiklik yapmak isteyebiliriz.
 - Örneğin bir değeri değiştirme, yeni değer ekleme, bir değeri silme gibi.
 - Her sistem bu fonksiyonları desteklemez.
 -

Fonksiyon	ISO C	POSIX.1	FreeBS 5.2.1	Linux 2.4.22	MAC OS X 10.3	Solaris 9
getenv	*	*	*	*	*	*
putenv		XSI	*	*	*	*
setenv		*	*	*	*	
unsetenv		*	*	*	*	
clearenv				*		

Çevre değişkenleri

```
#include <stdlib.h>
```

```
int putenv(char *str);
```

Dönüş: OK ise 0, hata ise 0 dışı değer.

```
int setenv(const char *name, const char *value, int rewrite);
```

```
int unsetenv(const char *name);
```

Dönüş: OK ise 0, hata ise -1.

- putenv fonksiyonu name=value şeklinde metni alır ve çevre değişkeni olarak ekler. Eğer değişken var ise eski değer silinir.
- setenv fonksiyonu bulunan bir değişkene değer atar. rewrite 0 dışı değer ise ve değer var ise eski değer silinir yeni değer yazılır. rewrite 0 ise ve değer var ise eski değer değiştirilmez.
- unsetenv fonksiyonu bir değişkeni siler. Değişken bulunmuyorsa hata dönmez.

getrlimit ve setrlimit fonksiyonları

- Her işleme ait kaynak limitleri mevcuttur. Bu limitlerin bazıları getrlimit ve setrlimit fonksiyonları ile öğrenilebilir ve değiştirilebilir.

```
#include <sys/resource.h>
```

```
int getrlimit(int resource, struct rlimit *rlptr);
```

```
int setrlimit(int resource, const struct rlimit *rlptr);
```

Dönüş: OK ise 0, hata ise 0 dışı değer.

- Bu fonksiyonlar bir kaynak tipi ve aşağıdaki yapıya işaretçi belirtir.

```
struct rlimit {
```

```
    rlim_t rlim_cur; /* anlık (soft) limit limit: şuanki  
    limit */
```

```
    rlim_t rlim_max; /* kesin (hard limit: rlim_cur  
    değerinin alabileceği en yüksek değer */
```

```
};
```

getrlimit ve setrlimit fonksiyonları

- Kaynak limitlerini değiştirmek için 3 kural vardır:
 - Bir işlem anlık limitini kesin limitinden az veya kesin limite eşit şekilde değiştirebilir.
 - Bir işlem kesin limitini anlık limiti kadar veya anlık limitin fazla olacak şekilde azaltabilir. Normal kullanıcılar için bu işlem geri alınamaz.
 - Sadece sistem yöneticisi kesin limiti artırabilir.
- Kaynak limit türlerini kitabınızdan inceleyebilirsiniz.
 - RLIMIT_AS, RLIMIT_CORE, RLIMIT_CPU, RLIMIT_FSIZE ...