BITWISE

BITWISE OPERATOR

& -> Bitwise AND

| -> Bitwise OR

<< -> Left shift

>> -> Right shift

RIGHT SHIFT

Bir bit bloğu üzerinde sağa kaydırma işlemi uyguladığımız zaman o blok üzerinde tüm bitler sağ pozisyona geçerler . En sağda ki bitler kaybolur, en soldan bir adet 0 eklenir .

Chapter1: 1011 -> 0101

Chapter2:

char c = 'a'; // 01100001 char c = 'a';

char b = c >> 1; // 00110000 c >>= 1;

Chapter3:

01000101 =

= 0x2^7 + 1x2^6 + 0x2^5 + 0x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 69 (ASCII karşılığı)

Not: Bir bit sağa kaydırırsak, onun integer değerini 2’ye bölmüş oluruz .

LEFT SHIFT

Bir bit bloğu üzerinde sola kaydırma işlemi uyguladığımız zaman o blok üzerindeki tüm bitler bir sol pozisyona geçerler . En soldaki bit kaybolur , en sağdan bit adet 0 eklenir .

Chapter1: 1011 -> 0110

Chapter2:

char c = 'a'; // 01100001 char c = 'a';

char b = c << 1; //11000010 c <<= 1;

Not: Bir bit sola kaydırdığımızda, onun integer değerini 2 ile çarpmış gibi oluruz .

AND OPERATOR (&)

İki bit bloğundaki tüm bitleri sırayla AND bağlacı ile sokmak için kullanılır .

0000 & 1111 = 0

1100 & 1010 = 0

0101 & 1011 = 1

Chapter2:

char a = 16; // a = 00010000

char b = 48; // b = 00110000

b &= a; // b = 00010000

OR OPERATOR (|)

İki bit bloğundaki tüm bitleri sırasıyla OR bağlacı ile işleme sokmak için kullanılır .

0000 | 1111 = 1

1100 | 1010 = 0

0101 | 1010 = 1

Chapter2:

char a = 16; // a = 00010000

char b = 32; // b = 00100000

b |= a; // b = 00110000

NOT:Tek bir opertör kullanıldığı zaman 8 bitin en sonuncu bitine bakaraktan karşılaştırma yapar .

KILL

kill sistem çağrısı , bir sürece veya bir süreç grubuna sinyal göndermekte kullanılabilir.

pid pozitif olduğunda, pid sürecine sig sinyali gönderilir.

kill komutu ile gönderilebilecek sinyaller kill -l ile listelenebilir.

Başarı durumunda sıfır döner. Hata oluşursa -1 döner .

PID

Bir Linux sisteminde her süreç oluşturulduğunda, onu diğer uygulamalara tanımlayan yeni bir numara verilir. Bu, işlem kimliği veya PID'dir ve çalışan işlemleri yönetmek için sistem genelinde kullanılır.Linux'un çalıştırdığı ilk sürece denir. systemd, PID 0 alır. Diğer tüm işlemler, alt öğeleri olarak oluşturulur.

Process, programın belleğe yüklenerek çalıştırılmasıyla ortaya çıkan yapıdır diyebiliriz. Her process in bir ID değeri vardır. Bu değer tekildir.

“ps veya ps aux” ile çalışan tüm programların pid öğrenme .

PAUSE

Sinyal gelene kadar beklemenin en basit yolu pause çağrısı yapmaktır. pause işlevi, bir sinyal eylemciyi çalıştırmak ya da süreci sonlandırmak gibi bir eylemi yerine getirmek üzere bir sinyalin gelmesini beklemek için kullanılır. Çağrı olaylarından biri gerçekleşene kadar çağrı sürecini veya çağrı dizisini uyku moduna geçirmek için kullanılır.

USLEEP

Burada belirttiğimiz bir mikro saniye kadar programın uyumasını (durmasını) sağlayan bir yapıdır . Bu sayede spam olarak gönderilen sinyalleri rahatça yakalayıp fonksiyona yönlendirmesini sağlar . Sleep ve Usleep arasında ki fark , sleepte parametre olarak saniye yer alırken usleepte mikro saniye cinsinden değer yazılır .

Link:<https://serdarkuzucu.com/bitwise-operatorler/>

Link:[Basit Sinyal İşleme (belgeler.org)](http://www.belgeler.org/glibc/glibc-Basic-Signal-Handling.html)

Link:[kill (belgeler.gen.tr)](http://belgeler.gen.tr/man/man2/man2-kill.html)

Link:[pause Kullanımı (belgeler.org)](http://www.belgeler.org/glibc/glibc-Using-Pause.html)