

Uygulama – Linux'ta proses yaratma

fork system çağrıları

- Mevcut bir prosten yeni bir proses oluşturmak için kullanılır
- Mevcut prosesi kopyalayarak yeni bir proses oluşturur.
- Yeni proses **cocuk** proses, orijinal proses ise **anne** proses
- Çocuk proses, anne prosesin tam bir kopyasıdır ve aynı program kodunu paylaşır, ancak kendi bağımsız bellek alanına sahiptir.
- Kullanılması: `pid_t pid = fork();`
- Bu çağrıdan sonra, iki proses de (anne ve çocuk) aynı kodu çalışmaya devam ederler ancak fork system çağrılarının dönüş değeri sayesinde ayırt edilebilirler:
 - anne proste, fork çağrıları çocuk prosesin pid'sini döndürür.
 - çocuk proste fork çağrısı sıfır döndürür.
 - fork başarısız olursa, -1 döndürür.

exec system çağrıları

- Mevcut proses görüntüsünü yeni bir proses görüntüsü ile değiştirir.
- Mevcut prosesin kodunu, verilerini ve yiğinını yeni bir programla değiştirir.
- Genellikle fork sistem çağrıları ile birlikte kullanılır, böylece çocuk proses başka bir program çalıştırır.
- exec ailesindeki birden fazla, farklı şekilde parametreler alan fonksiyonlar vardır.
Örneğin:
`int execlp(char * filename, char * arg0, char * arg1, ... ,
 char *argn, (char*) 0);`

Linux işletim sisteminde, **fork** ve **exec** sistem çağrıları farklı amaçlar için kullanılır:

1. **fork**: Bu sistem çağrıları, mevcut bir prosesi kopyalayarak yeni bir çocuk proses oluşturur. Çocuk proses, anne prosesin tam bir kopyasıdır ve aynı program kodunu paylaşır, ancak kendi bağımsız bellek alanına sahiptir. fork çağrıları, mevcut prosesin yürütülmesine devam ederken, yeni bir proses oluşturur.
2. **exec**: Bu sistem çağrıları ise mevcut prosesin görüntüsünü yeni bir proses görüntüsü ile değiştirir. Yani, exec çağrı yapıldığında, mevcut prosesin kodu, verileri ve yiğini yeni bir programla değiştirilir. exec fonksiyonları, belirtilen programı çalışmaya başlar ve mevcut prosesin yerine geçer, yani yeni bir proses yaratılmaz.