İşletim Sistemleri Uygulama 3

Linux'de `fork` ve `exec`

Bilgisayar Mühendisliği

İstanbul Teknik Üniversitesi
34469 Maslak, İstanbul

February 24, 2011

Bugün

İşletim Sistemleri Uygulama 3

`fork` sistem çağrısı

`exec` sistem çağrısı

fork kullanımı

- `int fork();`
- `fork()` bir defa çağrılır
- Ancak iki defa geri döner!!
  - Ana süreçte bir defa
  - Çocuk süreçte bir defa
- Ana ile çocuğu nasıl ayırt edebiliriz??
  - Çocukta döndürülen değer = 0
  - Anada döndürülen değer = çocuğun süreç tanımlayıcısı

Örnek Program

```
#include <stdio.h>                                                    1
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>

int main(void)                                                        6
{

        int f;

        f= fork();                                                    11

        if(f==-1){
                printf("Hata\n");
                exit(1);
        }                                                             16

        else if (f==0){
                printf("        Cocuk: Proses no: %d \n",getpid());
                sleep(1);
                printf("        Cocuk: Anne proses no: %d \n",getppid());  21
        }

        else{
                printf("Anne: Proses no: %d \n", getpid());
                printf("Anne: Cocuk proses no: %d \n", f);            26
                printf("Anne: Anne proses no: %d \n", getppid());
                wait(NULL);
                printf("Anne: Sonlaniyorum... \n");
                exit(0);
        }                                                             31

        return 0;
}
```

Örnek Program Çıktısı

```
ovatman@susa:/$ gcc forkOrnek.c
ovatman@susa:/$ ./a.out                          2
Anne: Proses no: 4319
Anne: Cocuk proses no: 4320
Anne: Anne proses no: 4284                       5
        Cocuk: Proses no: 4320
        Cocuk: Anne proses no: 4319
Anne: Sonlaniyorum...                            8
ovatman@susa:/$
```

exec kullanımı

```
int execlp(char * filename, char * arg0, char * arg1, . . . ,char
*argn,(char*)0);
```

- Unixin bir süreç içinden yeni bir program çalıştırmasının tek yoludur.
- PID değişmez
- Yeni bir süreç değildir!

exec kullanımı

Exec komutunun çağrılması için 6 yolu vardır.
`execl`, `execv`, `execle`, `execve`, `execlp`, `execvp`

- int execl (const char path, const char arg0, .../*, const char argn, (char *)0*/);
- int execv (const char path, char const argv);
- int execle (const char path, const char arg0, .../*, const char argn, (char *)0, const char envp[]*/);
- int execve (const char path, char const argv, char const envp);
- int execlp (const char file, const char arg0, .../*, const char argn, (char *)0*/);
- int execvp (const char file, char const argv);

Örnek Program

```
#include <stdio.h>                                                    1
#include <unistd.h>
#include <stdlib.h>

int f;
                                                                      6
int main(void)
{
        printf("\n Efendi calisiyor: PID:%d \n",getpid());

        f=fork();                                                     11

        if (f==0) {   //cocuk

                printf("\n Ben cocuk... PID: %d \n", getpid());
                execlp("./execSlave","./execSlave"                    16
                                ,"escrava","isaura",(char*)NULL);
        }
        else{   //anne
                exit(0);
        }                                                             21

        return 0;
}
```

Örnek Program

```
#include <stdio.h>                                                    1
#include <unistd.h>


int main(int argc, char* argv[])
{                                                                     6
        printf("\nKole calismaya basladi... PID: %d \n",getpid());
        printf("Programin adi:%s \n",argv[0]);
        printf("Programin 1. parametresi:%s \n",argv[1]);
        printf("Programin 2. parametresi:%s \n",argv[2]);
        return 0;                                                     11
}
```

Örnek Program Çıktısı

```
ovatman@susa:/$ gcc execMaster.c
ovatman@susa:/$ gcc execSlave.c -o execSlave
ovatman@susa:/$ ls
a.out
execSlave.c
execMaster.c
execSlave
ovatman@susa:/$ ./a.out

 Efendi calisiyor: PID:4499

 Ben cocuk... PID: 4500

Kole calismaya basladi... PID: 4500
Programin ad :./execSlave
Programin 1. parametresi:escrava
Programin 2. parametresi:isaura
ovatman@susa:/$
```