Örnek 1

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <stdlib.h>

int main(){

    pid\_t pid;

    pid=fork();

    if(pid<0) {

        printf("Error");

        exit(NULL);

    } else if (pid==0) {

        printf("Cocuk surec \n");

        execlp("ls","ls","-l",NULL);

    } else {

        printf("\nEbeveyn surec\n");

        wait(NULL);

    }

    return 0;

}

Örnek 2

#include <stdlib.h>

#include <stdio.h>

#include <sys/types.h> //pid\_t için

#include <unistd.h> //fork fonksiyonu için

#include <sys/wait.h> //wait komutu için

#include <time.h>

int main() {

    pid\_t cocuk\_pid;

    int c\_okunan;

    int durum;

    cocuk\_pid = fork(); //yeni oluşan sürecin pid'ini döndürür

    if (cocuk\_pid<0) { //fork başarısız

        //printf ("hata");

        perror("fork");

    } else if(cocuk\_pid==0) { //çocuğun tarafı

        printf("CHILD:Merhaba, ben çocuk süreç\n");

        printf("CHILD:Benim PID:%d\n",getpid()); //getpid() ile süreç kendi

id'sine ulaşır

        printf("CHILD:Ebeveynimin PID'si:%d\n",getppid()); //getppid() ile

süreç ebeveyninin id'sine ulaşır

        printf("CHILD:cocuk\_pid değişkeni:%d\n",cocuk\_pid);

        printf("CHILD:0-255 arası bir değer giriniz");

        scanf("%d",&c\_okunan);

        printf("CHILD:Çocuk çalışmasını tamamlıyor\n");

        exit(c\_okunan); //çocuk bir değer döndürüyor

    } else { //ebeveyn tarafı

        printf("PARENT:Merhaba ben ebeveyn süreç\n");

        printf("PARENT:Benim PID:%d\n",getpid());

        printf("PARENT:Çocuğumun PID'si:%d\n",cocuk\_pid);

        printf("PARENT:Çocuğun çalışması bekleniyor\n");

        wait(&durum);

        printf("PARENT:Çocucğun çıkış kodu:%d\n",WEXITSTATUS(durum));

        printf("Ebeveyn süreç çalışmasını tamamladı\n");

        exit(0);

    }

    return 0;

}

Örnek 3

#include<stdio.h>

#include<sys/wait.h>

#include<unistd.h>

int main()

{

    if (fork()== 0)

        printf("HC: hello from child\n");

    else

    {

        printf("HP: hello from parent\n");

        wait(NULL);

        printf("CT: child has terminated\n");

    }

    printf("Bye\n"); //hem çocuk hem ebeveyn çalıştırır

    return 0;

}