1.5. Algoritma geliştirebilmeli

2. Pipe

Pipe system cağrısı ebeyn-çocuk prosesler arası iletişimi sağlar. Pipe system çağırısı bir giriş/çıkış mekanızması oluşturur. Pipe çağrısı ile geri iki adet dosya tanımlayıcısı döner; fd[0],fd[1]. Giriş/çıkış işlemleri bu iki dosya tanımlayıcıları ile yapılır.

Genel kullanım:

Int pipe(int fd[2]);

Pipe çağırısına iki elemanlı tek boyutlu bir integer dizi gönderilir. Bu dizinin fd[0] okuma işlemi için kullanılacak dosyatanımlayıcıyı gösterir ve fd[1] yasmak için kullanılır. Pipe işlemi tek bir proses için kullanılmak mantıksızdır fakat çoklu proseslerin haberleşmesinde kullanılır.

Prog23 programını yazınız.

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
    int pipefd[2];
    int i;
    char s[1000];
    char *s2;
    if (pipe(pipefd) < 0) {
        perror("pipe");
        exit(1);
    }
    s2 = "fatih sultan mehmet";
    write(pipefd[1], s2, strlen(s2));
    i = read(pipefd[0], s, 1000);
    s[i] = '\0';
    printf("%d",pipefd[0]);
    printf("%d",pipefd[1]);
    printf("Pipe'tan %d byte Okundu:  '%s'\n", i, s);
    return 0;
}
```

```
insect@insect-Lenovo-G50-80: ~/OpSis/prog23
insect@insect-Lenovo-G50-80:~/OpSis/prog23$ ./prog23
Pipe'tan 19 byte Okundu:  'fatih sultan mehmet'
insect@insect-Lenovo-G50-80:~/OpSis/prog23$
```

Program 23'te görüldüğü gibi "pipe()" fonksiyonu pipefd dizisine dosya tanımlayıcılarını atmıştır. Pipefd[1]'e yazılan herhangi bir şey Pipefd[0]'dan okunabilmektedir. Programda bunun denemesini görmüş olduk. Eğer pipe ile atanan dosya tanımlayıcılarını yazdırırsanız (üste başka dosya tanımlayıcı atanmamışsa)

```
insect@insect-Lenovo-G50-80:~/OpSis/prog23$ ./prog23
Pipe'tan 19 byte Okundu:  'fatih sultan mehmet'
insect@insect-Lenovo-G50-80:~/OpSis/prog23$
```


görüldüğü gibi “pipe()” fonksiyonu pipefd dizisine dosya tanımlayıcılarını atmıştır. Pipefd[1]’e yazılan herhangi bir şey Pipefd[0]’dan okunabilmektedir. Programda bunun denemesini görmüş olduk. Eğer pipe ile atanan dosya tanımlayıcılarını yazdırırsanız (üste başka dosya tanımlayıcı atanmamışsa) alacağınız değerler 3 ve 4’tür. Bundan onceki dosya tanımlayıcı değerleri standart input(0), standart output(1), standart error(2)’dir.

Prog24 programını yazınız.

```c
#include <stdio.h>
int main(int argc, char *argv[])
{
    int pipefd[2];
    int pid;
    int i, line;
    char s[1000];

    if (pipe(pipefd) < 0) {
        perror("pipe");
        exit(1);
    }
    pid = fork();

    if (pid > 0) {
        while(fgets(s, 1000, stdin) != NULL) {
            write(pipefd[1], s, strlen(s));
        }
        close(pipefd[1]);
    }
    else {
        i = 0;
        line = 1;
        while(read(pipefd[0], s+i, 1) == 1) {
            if (s[i] == '\n') {
                s[i] = '\0';
                printf("%6d  %s\n", line, s);
                line++;
                i = 0;
            }
            else {
                i++;
            }
        }
    }
    return 0;
}
```

```
insect@insect-Lenovo-G50-80: ~/OpSis/prog24
insect@insect-Lenovo-G50-80:~/OpSis/prog24$ ./prog24
fatih
     1  fatih
sultan
     2  sultan
mehmet
     3  mehmet
vakif
     4  vakif
Universitesi
     5  Universitesi
```

Programda ebeyn proses pipe’a yazmakta ve diğer çocuk proses ise okuma işlemi gerçekleştirmektedir.

```c
char *convert(int number)
{

main()
{
    int pipefd[2];
    int pid;
    int i, line;
    char s[1000];
    int c;

    if (pipe(pipefd) < 0) {
        perror("pipe");
        exit(1);
    }
    char *newargv[2];
    newargv[0] =  "convert(pipefd[0])";
    newargv[1] =  "convert(pipefd[1])";
    newargv[2] = NULL;
    pid = fork();

      if (pid == 0) {
          c = execve("prog25_2", newargv);
          perror("");
          close(pipefd[1]);
      }
      else {
        wait(&c);
        printf("Ana program:getpid: %d  getpppid: %d\n", getpid(), getppid());
        i = 0;
        line = 1;
        while(read(pipefd[0], s+i, 1) == 1) {
          if (s[i] == '\n') {
            s[i] = '\0';
            printf("%6d  %s\n", line, s);
            line++;
            i = 0;
          } else {
            i++;
          }
        }
    }
    return 0;
}
```

```
#include <stdio.h>
                                                    rnums)
int main(int argc, char *argv[])
{

    char s[1000];
    printf("Alt program:getpid: %d  getpppid: %d\n", getpid(), getppid());
    fgets(s, 1000, stdin);
    [redacted]s, strlen(s));
    return 0;
}
```

```
insect@insect-Lenovo-G50-80: ~/OpSis/prog25
insect@insect-Lenovo-G50-80:~/OpSis/prog25$ ./prog25
Alt program:getpid: 13928  getpppid: 13927
fatih sultan mehmet vakif Universitesi
Ana program:getpid: 13927  getpppid: 7570
     1  fatih sultan mehmet vakif Universitesi
```

3. Kaynaklar

1. **Wikipedia.** ***wikipedia.org.*** **[Online] https://en.wikipedia.org/wiki/Operating_system.**

2. **Computerhope.** ***www.computerhope.com.*** **[Online] http://www.computerhope.com/jargon/o/os.htm.**

3. **https://atessinan.wordpress.com/2012/03/03/sistem-programlama-3-write-read/**