

SİSTEM PROGRAMLAMA

Deney#5 : UNIX 'te IPC mekanizması olarak Pipe kullanımı

1. isimsiz_pipe.c adında dosya oluşturup içinde kullanıcı adı kısmını kendi adım ile değiştirdim.

```
isimsiz_pipe.c:31:1: warning: implicit declaration of function 'wait' is invalid [-Wimplicit-function-declaration]
31 | wait (0);
    | ^~~~~
ilhan@linux-mint:~/sistemp/odev3$ ./a.out
adm:x:4:syslog,ilhan
cdrom:x:24:ilhan
sudo:x:27:ilhan
dip:x:30:ilhan
plugdev:x:46:ilhan
lpadmin:x:115:ilhan
ilhan:x:1000:
sambashare:x:135:ilhan
ilhan@linux-mint:~/sistemp/odev3$
```

2. % getent group : sistemdeki tüm grupları listeliyor; ilhan kullanıcısı 7 farklı grubun üyesi olarak görünüyor.

```
crontab:x:104:
messagebus:x:105:
systemd-timesync:x:106:
input:x:107:
sgx:x:108:
kvm:x:109:
render:x:110:
syslog:x:111:
tss:x:112:
rtkit:x:113:
systemd-coredump:x:114:
lpadmin:x:115:ilhan
bluetooth:x:116:
ssl-cert:x:117:
netdev:x:118:
uudd:x:119:
lightdm:x:120:
nopasswdlogin:x:121:
tcpdump:x:122:
_ssh:x:123:
plocate:x:124:
avahi-autoipd:x:125:
nm-openvpn:x:126:
geoclue:x:127:
pulse:x:128:
pulse-access:x:129:
scanner:x:130:saned
_flatpak:x:131:
avahi:x:132:
saned:x:133:
colord:x:134:
ilhan:x:1000:
sambashare:x:135:ilhan
fwupd-refresh:x:136:
ilhan@linux-mint:~/sistemp/odev3$
```

```
bin:x:2:
sys:x:3:
adm:x:4:syslog,ilhan
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:ilhan
floppy:x:25:
tape:x:26:
sudo:x:27:ilhan
audio:x:29:pulse
dip:x:30:ilhan
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:ilhan
staff:x:50:
games:x:60:
```

% getent group | grep ilhan: bu komut ile sistemdeki tüm gruplarda parametre olarak verilen kullanıcı adının (ilhan) geçtiği grupları listeliyor.

Programımız ise ilhan kullanıcısının dahil olduğu grupları listeliyor

```
ilhan@linux-mint:~/sistemp/odev3$ ./a.out
adm:x:4:syslog,ilhan
cdrom:x:24:ilhan
sudo:x:27:ilhan
dip:x:30:ilhan
plugdev:x:46:ilhan
lpadmin:x:115:ilhan
ilhan:x:1000:
sambashare:x:135:ilhan
ilhan@linux-mint:~/sistemp/odev3$ getent group|grep ilhan
adm:x:4:syslog,ilhan
cdrom:x:24:ilhan
sudo:x:27:ilhan
dip:x:30:ilhan
plugdev:x:46:ilhan
lpadmin:x:115:ilhan
ilhan:x:1000:
sambashare:x:135:ilhan
ilhan@linux-mint:~/sistemp/odev3$
```

3. Aşağıda değiştirdiğim kod blokları görülüyor. execlp ("getent", "getent", "group", 0) devre dışı bırakıldı. Pipe ın yazma ucuna write([1], "mesaj\n", mesajBoydu) şeklinde istenen kelimeler gönderildi.

```
ilhan@linux-mint:~/sistemp/odev3$ make clean
rm a.out
ilhan@linux-mint:~/sistemp/odev3$ make
cc isimsiz_pipe.c
isimsiz_pipe.c:3:1: warning: return type defaults to 'int'
3 | main ()
  | ~~~~~
isimsiz_pipe.c: In function 'main':
isimsiz_pipe.c:12:1: warning: implicit declaration of function 'exit'
12 | exit (1);
   | ~~~~~
isimsiz_pipe.c:3:1: note: include '<stdlib.h>' or provide a declaration for 'exit'
2 | #include <unistd.h>
+++ |+#include <stdlib.h>
3 | main ()
isimsiz_pipe.c:12:1: warning: incompatible implicit declaration of function 'exit'
12 | exit (1);
   | ~~~~~
isimsiz_pipe.c:12:1: note: include '<stdlib.h>' or provide a declaration for 'exit'
isimsiz_pipe.c:45:1: warning: implicit declaration of function 'wait'
45 | wait (0);
   | ~~~~~
ilhan@linux-mint:~/sistemp/odev3$ ./a.out
This
is
a
message
from
sending
process
ilhan@linux-mint:~/sistemp/odev3$
```

```

pid = fork (); /* Parent: ilk çocuk prosesi oluşturdu */
if (pid == 0) /* İlk çocuk prosesi burada başlatılıyor */
{
    dup2 (fd[0],0); /* Standart giriş, pipe ın okuma ucundan alınacak */
    close (fd[1]); /* Pipe yazma ucu bu işlem için kapatılacak, kullanılmayacak */
    while(read(fd[0],buf,sizeof(buf))>0)
    {read(fd[0],buf,sizeof(buf));
     write(1,buf,sizeof(buf));
    }
    close(fd[0]);
    //execlp ("grep", "grep", "ilhan", 0); /* "grep" komutu pipten veri oku */
}
else /* Parent prosesi devam ediyor*/
pid = fork (); /* Parent: ikinci çocuk prosesi oluşturdu */
if (pid == 0) /* ikinci çocuk prosesi burada başlatılıyor*/
{dup2 (fd[1],1); /* Standard çıkış pipe ın yazma ucuna konacak */
 close (fd[0]); /* Bu işlem süresince okuma ucu kapalı olacak, kullanılmayacak */
 write(fd[1],"This\n",5);
 write(fd[1],"is\n",3);
 write(fd[1],"a\n",2);
 write(fd[1],"message\n",8);
 write(fd[1],"from\n",5);
 write(fd[1],"sending\n",8);
 write(fd[1],"process\n",8);
 close(fd[1]);
 //execlp ("getent", "getent", "group", 0); /* "getent" komutu pipe veri oku */
}
```

4. `execlp ("grep", "grep", "ilhan", 0)` komutu devre dışı bırakıldı. `read(fd[0],buf,mesajBoyuu)` komutu ile mesajlar pipeten okunarak `write(1,buf,mesajBoyuu)` komutu ile ekrana yazıldı.
5. İstenen şekilde 2 yazıcı bir de okuyucu proses yarattım. Çıktıda da görüldüğü şekilde yazıcılar farklı mesajları pipe a yazdılar. Okuyucu proses pipeden mesajları karışmadan yazdı.(en azından bu mesajlarda)

```
2.
ç2y2
yaz2
ilhan@linux-mint:~/sistemp/odev3$ ./a.out
1.
ç1y1
yaz1
2.
ç2y2
yaz2
ilhan@linux-mint:~/sistemp/odev3$ ./a.out
1.
ç1y1
yaz1
2.
ç2y2
yaz2
ilhan@linux-mint:~/sistemp/odev3$ ./a.out
1.
ç1y1
yaz1
2.
ç2y2
yaz2
ilhan@linux-mint:~/sistemp/odev3$ ./a.out
```

```
*isimsiz_pipe.c x
}
pid = fork (); /* Parent: 1. çocuk prosesi
if (pid == 0) /* ikinci çocuk proses burda
{dup2 (fd[1],1); /* Standard çıkış pipe in
close (fd[0]); /* Bu işlem süresince okuma
write(fd[1],"1.\n",3);
write(fd[1],"ç1\n",3);
write(fd[1],"y1\n",3);
write(fd[1],"yaz1\n",5);
close(fd[1]);
}
else /* Parent proses devam ediyor*/
pid = fork (); /* Parent: 2. çocuk prosesi
if (pid == 0) /* üçüncü çocuk proses burda
{dup2 (fd[1],1); /* Standard çıkış pipe in
close (fd[0]); /* Bu işlem süresince okuma
write(fd[1],"2.\n",3);
write(fd[1],"ç2\n",3);
write(fd[1],"y2\n",3);
write(fd[1],"yaz2\n",5);
close(fd[1]);
}
else
pid = fork (); /* Parent: 3. çocuk prosesi
if (pid == 0) /* İlk çocuk proses burda ba;
{
dup2 (fd[0],0); /* Standart giriş, pipe in
close (fd[1]); /* Pipe yazma ucu bu işlem :
while(read(fd[0],buf,sizeof(buf))>0)
{read(fd[0],buf,sizeof(buf));
write(1,buf,sizeof(buf));
}
close(fd[0]);
}
```

6. Farklı proseslerin kullandığı iki ayrı pipe yaratıldı ve echo işlemi sağlandı

```
ilhan@linux-mint:~/sistemp/odev3$ ./a.out
bu
mesaj
geri
donmeli
```

```

if (pipe(f2) < 0)
{perror ("Pipe oluşturma hatası");
exit (1);
}

pid = fork ();
if (pid == 0)
{dup2 (f2[1],1);
close (f2[0]);
write(f2[1],"bu\n",3);
write(f2[1],"mesaj\n",6);
write(f2[1],"geri\n",5);
write(f2[1],"donmeli\n",8);
close(f2[1]);

dup2 (f1[0],0);
close (f1[1]);
while(read(f1[0],buf,sizeof(buf))>0)
{read(f1[0],buf,sizeof(buf));
write(1,buf,sizeof(buf));
}
close(f1[0]);
}
else /* Parent proses devam ediyor*/
pid = fork ();
if (pid == 0)
{
dup2(f2[0],0);
close (f2[1]);
dup2(f1[1],1);
close (f1[0]);

while(read(f2[0],buf,sizeof(buf))>0)
{read(f2[0],buf,sizeof(buf));
write(f1[1],buf,sizeof(buf));
}
close(f2[0]);
close(f1[1]);
}

else /* Ana işlem pipe uçlarını kendisi için kapatıp
{
close (f1[0]);
close (f1[1]);
close (f2[0]);
close (f2[1]);
wait (0);
wait (0);
}
}

```

2.Bölüm

1. Kodlar derlendi.
2. Yaratılma “boru” pipe görülüyor.

```
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./snc
SUNUCU BASLAR...
mkfifo problem in server: File exists
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ls
boru client.c fifo-cln.c fifo-srv.c ist istemci.c makefile server.c snc sunucu.c
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$
```



```
istemci.c:22:1: warning: implicit declaration of function 'close' [-Wimplicit-fun
22 | close(fd);
    | ^~~~~
istemci.c:23:1: warning: incompatible implicit declaration of built-in function
laration-mismatch]
23 | exit(0);
    | ^~~~
istemci.c:23:1: note: include '<stdlib.h>' or provide a declaration of 'exit'
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ls
client.c fifo-cln.c fifo-srv.c ist istemci.c makefile server.c snc sunuc
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./ist
Istemci BASLAR...
open problem in client: No such file or directory
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./snc
SUNUCU BASLAR...
meraba^C
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./ist
Istemci BASLAR...
DDD
```

3-4. Sunucu ve istemciyi aynı konsolda açtım ve istemciyi açtıktan sonra bir mesaj yazıp Ctrl+D yaptım ancak bir şey gözlemlemedim. Sonra sunucu ve istemciyi ayrı konsollarda açtım. Önce sunucuyu sonra istemciyi açıp istemciden mesaj yolladım ancak yine sunucuda mesajı göremedim. Kodlar şu şekilde ama sonuç yok.

```

sunucu.c x istemci.c x
/* ISTEMCI PROGRAM*/
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#define boru

int main(void)
{
    int n, fd;
    char buf[1024];
    printf ("ISTEMCI BASLAR...\n");

    if ((fd = open("boru", O_WRONLY)) < 0) {
        perror("open problem in client");
        exit(1);
    }

    while ((n = read(0, buf, sizeof(buf))) > 0)
        write(fd, buf, n);

    printf ("ISTEMCI SONLANDI...\n");
    close(fd);
    exit(0);
}

```

```

sunucu.c x istemci.c x
/* SUNUCU PROGRAM */
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#define boru
/*myfifo burada pipe özel bir ad verin */
int main(void)
{
    int n, fd;
    char buf[1024]; /* okuma ve yazma için kullanılacak buffer */
    printf ("SUNUCU BASLAR...\n");

    unlink(boru);

    if (mkfifo("boru", 0666) < 0) {
        perror("mkfifo problem in server");
        exit(1);
    }

    //if (chmod(boru, 0666) < 0) perror("chmod problem in server"); exit(1);

    fd = open("boru", 0666);
    if (fd < 0) {
        perror("open problem in server");
        exit(1);
    }

    while ((n = read(fd, buf, sizeof(buf))) > 0)
        write(1, buf, n);

    close(fd);
    printf ("SUNUCU SONLANDI...\n");
    exit(0);
}

```

6. Pipe isimlerini parametre olarak dışardan girince program doğru çalıştı. İstemciden gönderilen “adaad” mesajı serverın çalıştığı konol ekranına yollandı.

```

boru ~$ cd ~/sistemp/odev3/sunucu_istemci
client.c fifo-cln.c istemci.c server.c sunucu.c
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./cln boru
adaad

```

```

ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ls
client.c  cln  fifo-cln.c  fifo-srv.c  istemci.c  makefile  server.c  srv  sunu
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./srv boru
ls
sdsdfsdf
^C
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./srv boru
^C
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ^C
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./srv boru
adaad

```


7. Hazır olarak verilen server.c ve client.c dosyaları p1 ve p2 pipe isimleri parametre olarak çalıştırıldı. Sunucu ve İstemci başarılı şekilde başladı . İstemci başlar başlamaz sunucu bunu teyit etti ve mesaj bekledi. İstemci tarafındaki konsoldan bir mesaj yazarak Ctrl+D tuşu ile göndermek istedim ancak mesaj gitmedi. Ekran görüntüleri aşağıda verilmiştir.

```
ilhan@linux-mint: ~/sistemp/odev3/sunucu_istemci
Dosya Düzenle Görünüm Ara Uçbirim Yardım
server.c:56:1: warning: implicit declaration of function 'close'; did you mean '
close'? [-Wimplicit-function-declaration]
  56 | close (fd1);
      | ^~~~~
server.c:62:42: warning: incompatible implicit declaration of built-in function
exit' [-Wbuiltin-declaration-mismatch]
   62 | { perror("OPEN ERROR FOR SECOND PIPE"); exit(1); }
      | ^~~~~
server.c:62:42: note: include '<stdlib.h>' or provide a declaration of 'exit'
server.c:70:1: warning: incompatible implicit declaration of built-in function '
xit' [-Wbuiltin-declaration-mismatch]
   70 | exit(0);
      | ^~~~~
server.c:70:1: note: include '<stdlib.h>' or provide a declaration of 'exit'
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./srv p1 p2
SERVER STARTS...
FIRST PIPE IS CREATED
SECOND PIPE IS CREATED
TRYING TO OPEN THE FIRST PIPE...
THE FIRST PIPE IS OPENED FOR READING
TRYING TO READ A REQUEST FROM CLIENT...

ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./srv p1 p2
SERVER STARTS...
FIRST PIPE IS CREATED
SECOND PIPE IS CREATED
TRYING TO OPEN THE FIRST PIPE...
merhaba^C
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ls
ls: komut bulunamadı
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ls
client.c fifo-cln.c istemci.c p1 server.c sunucu.c
cln fifo-srv.c makefile p2 srv
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ rm p1 p2
ilhan@linux-mint:~/sistemp/odev3/sunucu_istemci$ ./cln p1 p2
CLIENT STARTED...
A PIPE IS OPENED FOR WRITING
A REQUEST IS SENT TO SERVER...
merhaba
^C
```

```
printf("TRYING TO READ A REQUEST FROM CLIENT...\n");

while(n1=read(fd1,buf1,sizeof(buf1))>0)
write(1,buf1,n1);
close (fd1);
printf("A REQUEST IS RECEIVED FROM CLIENT...\n");

printf("\nTRYING TO OPEN THE SECOND PIPE...\n");
/* Working with the second pipe */
if(fd2=open(argv[2],O_WRONLY)<0)
{ perror("OPEN ERROR FOR SECOND PIPE"); exit(1); }
printf("SECOND PIPE IS OPENED FOR WRITING\n");

write(fd2,"THIS IS A REPLY FROM SERVER\n",28);
printf("A REPLY MESSAGE SENT TO CLIENT\n");
close(fd2);
```