

Write two LINUX programs that will exchange messages and quit. These programs will both be based on the program you already created. Call these programs the “Left” program and the “Right” program. The “Left” program must create a shared memory segment of size 1000, and copy a string into it. The “Right” program must attach the shared memory segment, read the string it contains, print it, send a reply, and exit. When the “Left” program receives the reply, it must read it, print it, destroy the memory segment and exit. The code to attach an existing shared segment to a program is identical to that which creates the shared segment: as you may recall:

```
int MyLastFour=1111, SegSize=1000;
int MyShmid = shmget(MyLastFour, SegSize, IPC_CREAT|0x1c0);
char * MyAddr = shmat(MyShmid, NULL, 0);
```

Use the first character of the shared segment to indicate whose turn it is to write into the shared segment. The “Left” program should store a null character in the first character like this:

```
*MyAddr = '\0';
```

The message from the Left program should be “Hello From Left”. Copy this into the shared segment, but don’t overwrite the first byte.

```
strcpy(MyAddr+1, “Hello From Left”);
```

After copying in the message, put the character ‘A’ in byte 1 to signify that the message has been sent. The Right program will wait for the letter A using the following loop.

```
while (MyAddr != ‘A’)
{
    sleep(1);
}
```

This is a “not too busy wait” because the sleep function causes the program to wait for 1 second, while giving up control of the CPU. After sending the message, the Left program should wait for a ‘B’ to appear in byte 1 of the shared memory segment. The Right program will print the message from the left program and send the message “Goodbye from Right”. Remember to copy the message before inserting the ‘B’. The Left program will read this message and print it. The Right program should disconnect from the shared memory segment, but should not destroy it. The Left program should do both.

Turn in: Floppy Disk or CD ROM with source code, and a printout of your source code.

This protocol could be used to exchange messages between two processes, but it might not work under all circumstances. Why not?