

## Lab 4: Redirecting Pipes



### **Exercise:**

Have an inactivity detector.

Your program should read words (using `scanf`) (!) from the keyboard and print the text right away, but with exclamation marks at the beginning and the end of the word.

If the user is not entering anything for 10 seconds (to make it easier, at least 10 seconds), overwrite `stdin` and print out “no activity detected!”. Go back into read mode and wait for another 10 seconds and so on.

### **HowTo:**

Fork into a parent and a child.

Have the parent in an infinite loop of reading, text updating and printing.

Have a mechanic (shared mem?) to tell the child process if an activity happened.

The child process is in an infinite loop as well, waiting for 10 seconds. If no activity was reported by the parent process, overwrite `stdin` and print out “no activity detected”, followed by restoring `stdin` for letting the parent process read normally from the keyboard again.

If you are entering “quit” the program should end!

The text above contains ALL necessary information regarding the program. Read and understand it carefully.

Use functions. E.g. for updating the text.

Use pipes

### **How it looks:**

```
>Hello World
!Hello World!
>Everything seems to work fine
!Everything seems to work fine!
[more than 10s pause]
Inactivity detected!
>Many students clearly haven't read my very first email where I stated not to be called
professor but Christian
! Many students clearly haven't read my very first email where I stated not to be called
professor but Christian!
>quit
```

### **What you need:**

```
getpid()
signal()
kill()
fork()
wait()
dup()
dup2()
sleep()
mmap()
munmap()
read()
write()
```

### **Bonus:**

100 respect points for whoever knows the artist of the pipe-illustration above.