For receiver.c :

The basic idea is that it can create a shared memory, attach to that memory and then put strings to the memory. If input “END”, then just end the program.

1. Create a shared memory
   shmid = shmget(MYKEY,BUF_SIZE,IPC_CREAT)
   Here, the key is used to identity shared memory.
2. Attach to the memory
   shmptr =shmat(shmid,0,0)
3. Get input to the memory address
   scanf("%s",shmptr)

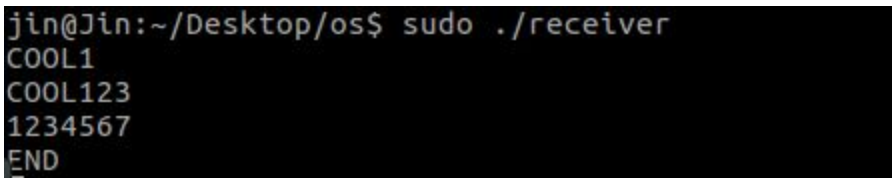
For processor.c:

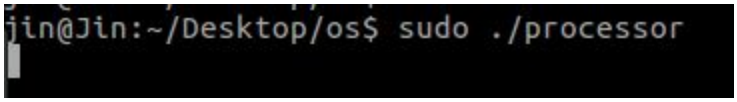
The basic idea is that it should create a shared memory, attach to that memory through the key, and read data from that memory. I will make a copy of previous string and compare it to the current string. If they are the same, just sleep 1 second.

1. Create a shared memory with the same key.
   shmid = shmget(MYKEY,BUF_SIZE,IPC_CREAT)
2. Attach to that memory
   shmptr = shmat(shmid,0,0)
3. Then, open the output file, secrets.out.
   ```
   if((fp = fopen("secrets.out","ab+")) == NULL){
       printf("open secrets.out failed!");
       exit(0);
   }
   ```
4. Make a copy of current string which is used to compare in the future.
   ```
   char tmp[1024];
   strcpy(tmp,shmptr);
   ```
5. Check the string whether has “COOL”, count digits and write into file. If the input contains “END”, just end the programm.
   ```
   while(1){
       // it means the string in memory doesn't change. So just sleep 1s.
       if(strcmp(tmp,shmptr) == 0){
           sleep(1);
       }
       else{
           // copy the current string
           strcpy(tmp,shmptr);
           // if it has “COOL”, count digits and write to file
   ```

```
                if(strstr(tmp,"COOL") != NULL){

                        int count = 0;
                        for(int i =0; tmp[i]!='\0';i++){
                                if(tmp[i] - '0' >= 0 && tmp[i] - '0' <= 9){
                                        count++;
                                }
                        }
                        char digit[10];
                        fputs(shmptr,fp);
                        fputs("  number of digits: ",fp);
                        sprintf(digit, "%d", count);
                        fputs(digit,fp);
                        fputs("\n",fp);

                }
                if(strstr(tmp,"END") != NULL){
                        fclose(fp);
                        break;
                }

        }
}
```

Result:






```
COOL1  number of digits: 1
COOL123  number of digits: 3
```