

Week 9 in-class exercises. Pipe

In the last worksheet, we wrote a program that forked one child for each command line argument. The child computes the length of the command line argument and exits with that integer as the return value. The parent sums these return codes and reports the total length of all the command line arguments together. For this worksheet, we will do the same program except that each child will communicate the length to the parent through a pipe.

```
int main (int argc, char **argv) {  
    // Declare any variables you need
```

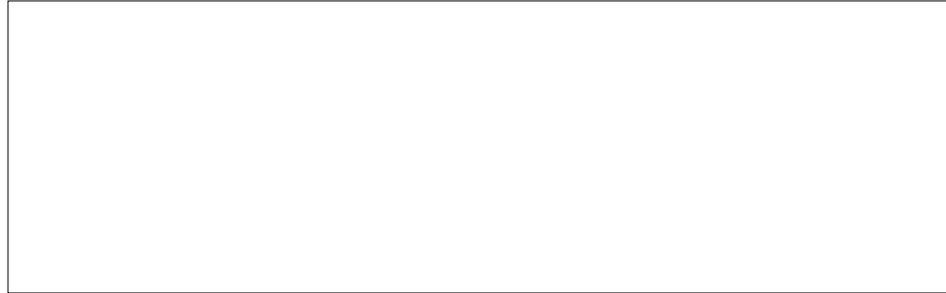
```
    // Write the code to loop over the command line arguments  
    for (int i = 0; i < argc - 1; i++) {  
        // Call pipe before we fork
```

```
        // Call fork  
        int result = fork();
```

```
        if (result == 0) {  
            printf("Inside child: %d, %d\n",  
                pipe_fd[i][0], pipe_fd[i][1]);  
            // Child only writes to the pipe, so close reading end
```

```
        // Before we forked, parent had open the reading ends to  
        // all previously forked children; so close those.
```

```
//Now do the work -  
//and write the value in binary to the pipe
```



```
// Close the pipe since we are done with it.  
close(pipe_fd[i][1]);  
exit(0);
```

```
}  
else {  
    printf("Inside parent: %d, %d\n",  
           pipe_fd[i][0], pipe_fd[i][1]);  
    // In the parent, but before the next loop iteration,  
    // close the writing end of the pipe
```



```
}  
}
```

```
// Only the parent gets here  
//reads from each pipe and computes the sum
```



```
return 0;
```

```
}
```