CSE344 Homework-2 Report

Student Name: Ahmet Hakan Sevinç

Student NO: 200104004047

15.04.2024

Executing the program simply by using make commands:

- make
- make run
- make clean

Output Examples:

```
hakan@DESKTOP-52VKK0J:~/hw2$ ls
main.c  makefile
hakan@DESKTOP-52VKK0J:~/hw2$ make
gcc  -c main.c -o main.o
gcc  main.o -o main
hakan@DESKTOP-52VKK0J:~/hw2$ make run
./main
Enter number of elements in array:5
Choose the array you want to craete.
Press 1 for Array: 1,2,3,...n
Press 2 for Random Array.
1
Array: 1 2 3 4 5
Proceeding...
Proceeding...
Proceeding...
Proceeding...
Proceeding...
Sum child-2: 135
Child 20858 terminated with status 0
Child 20859 terminated with status 0
All children terminated
hakan@DESKTOP-52VKK0J:~/hw2$ ls
fifo1  fifo2  main  main.c  main.o  makefile
hakan@DESKTOP-52VKK0J:~/hw2$ make clean
rm -f main.o main
find -type p -exec rm -f {} \;
hakan@DESKTOP-52VKK0J:~/hw2$ ls
main.c  makefile
hakan@DESKTOP-52VKK0J:~/hw2$
```

```
hakan@DESKTOP-52VKK0J:~/hw2$ ls
main.c  makefile
hakan@DESKTOP-52VKK0J:~/hw2$ make
gcc  -c main.c -o main.o
gcc  main.o -o main
hakan@DESKTOP-52VKK0J:~/hw2$ make run
./main
Enter number of elements in array:4
Choose the array you want to craete.
Press 1 for Array: 1,2,3,...n
Press 2 for Random Array.
2
Array: 8 6 10 9
Proceeding...
Proceeding...
Proceeding...
Proceeding...
Proceeding...
Sum child-2: 4353
Child 24619 terminated with status 0
Child 24620 terminated with status 0
All children terminated
hakan@DESKTOP-52VKK0J:~/hw2$ ls
fifo1  fifo2  main  main.c  main.o  makefile
hakan@DESKTOP-52VKK0J:~/hw2$ make clean
rm -f main.o main
find -type p -exec rm -f {} \;
hakan@DESKTOP-52VKK0J:~/hw2$ ls
main.c  makefile
```

```
hakan@DESKTOP-52VKK0J:~/hw2$ ls
main.c  makefile
hakan@DESKTOP-52VKK0J:~/hw2$ make
gcc  -c main.c -o main.o
gcc  main.o -o main
hakan@DESKTOP-52VKK0J:~/hw2$ make run
./main
Enter number of elements in array:3
Choose the array you want to craete.
Press 1 for Array: 1,2,3,...n
Press 2 for Random Array.
2
Array: 4 7 2
Proceeding...
Proceeding...
Proceeding...
Proceeding...
Proceeding...
Sum child-2: 69
Child 23933 terminated with status 0
Child 23934 terminated with status 0
All children terminated
hakan@DESKTOP-52VKK0J:~/hw2$ ls
fifo1  fifo2  main  main.c  main.o  makefile
hakan@DESKTOP-52VKK0J:~/hw2$ make clean
rm -f main.o main
find -type p -exec rm -f {} \;
hakan@DESKTOP-52VKK0J:~/hw2$ ls
main.c  makefile
hakan@DESKTOP-52VKK0J:~/hw2$
```

```
hakan@DESKTOP-52VKK0J:~/hw2$ ls
main.c  makefile
hakan@DESKTOP-52VKK0J:~/hw2$ make
gcc  -c main.c -o main.o
gcc  main.o -o main
hakan@DESKTOP-52VKK0J:~/hw2$ make run
./main
Enter number of elements in array:10
Choose the array you want to craete.
Press 1 for Array: 1,2,3,...n
Press 2 for Random Array.
1
Array: 1 2 3 4 5 6 7 8 9 10
Proceeding...
Proceeding...
Proceeding...
Proceeding...
Proceeding...
Sum child-2: 3628855
Child 22599 terminated with status 0
Child 22600 terminated with status 0
All children terminated
hakan@DESKTOP-52VKK0J:~/hw2$ ls
fifo1  fifo2  main  main.c  main.o  makefile
hakan@DESKTOP-52VKK0J:~/hw2$ make clean
rm -f main.o main
find -type p -exec rm -f {} \;
hakan@DESKTOP-52VKK0J:~/hw2$ ls
main.c  makefile
hakan@DESKTOP-52VKK0J:~/hw2$
```

How is synchronization handled?

The Program uses FIFO and sleeps. I used sleep 10 seconds after open files. Because If files don't be opened, parent will wait to open to write. There are two writers to second fifo and one reader which is child-2. Reader can read before one of writers write the fifo. For this scenario I try to read after and after until read becomes success.

Error Handling:

The program writes the messages to stderr, if error occurs.

Zombie protection and printing exit status:

The parent process wait until child processes are dead. It uses waitpid() function to perform finding exit status by setting options to WHOHANG.