

Program Description

- Starter receives the filename through the command line argument.
- Starter then creates a pipe and checks for successful creation.
- Pass the pipe reference to **Reader** for maintaining a running sum of the inputs.
- **Reader** writes the sum to the pipe using the provided reference. (only the write end is required)

Program Description

- The control is passed back to the **Starter** file where we reads contents of the pipe into a char array.
- **Starter** finds the maximum prime number from an integer value of the char array.
- Starter creates three shared memory segments, for Lucas, HarmonicSeries, and HexagonalSeries.
- Further, we print the name and the file descriptor of the shared memory.

Program Description

- Fork the Lucas, HarmonicSeries, and HexagonalSeries programs, and pass the name of the corresponding shared memory segment.
- The Lucas, HarmonicSeries, and HexagonalSeries write the last value calculated to the shared memory segment.
- The three child processes must run concurrently and not sequentially.
- **Starter** waits for all the child processes to complete and then prints the return value from the shared memory.
- Finally, unlink the shared memory.

CS 370 - Operating Systems - Spring 2022

Run Processes Concurrently

- In Assignment 2, the wait condition for the child was written before the parent process forked the next child.
- This leads to linear/sequential execution. However, for this Assignment, we need to execute the programs concurrently.
- Hence, the **Starter** must fork all the child processes and then use the wait() command for each of them.

Function Description

- pipe()
- shm_open()
- ftruncate()

- mmap()
- shm_unlink()
- sprintf()

pipe()

Syntax: int pipe(int pipefd[2]);

Arguments: pipefd[2] is the array to represent two ends of the pipe. Each end is a file descriptor (FD).

Example: int pipefds[2];

int result_pipe = pipe(pipefds);

shm_open()

Syntax:int shm_open(const char *name, int oflag, mode_t mode);Arguments:name: name of the memory segment
oflag: can take the following values: O_RDONLY, O_RDWR,
O_CREAT, O_EXCL, O_TRUNC
mode: permissions in the form 0666Example:char shm_Name[15] = "Shared_Mem0";
int shm_fd = shm_open(shm_Name, O_CREAT | O_RDWR,
0666);

CS 370 - Operating Systems - Spring 2022

ftruncate()

Syntax: int ftruncate(int fd, off_t length);

Arguments: fd: is the file descriptor

length: is the desired size of the memory segment. (Will be initialized to 0)

Example: int result = ftruncate(fd, 1234);

mmap()

Syntax:	void *mmap(void *addr, size_t length, int prot, int flags, int fd, off_t offset);
Arguments:	addr: beginning address of the memory object
	length: length of the memory object in bytes
	prot : protection of the pages (PROT_EXEC, PROT_READ, PROT_WRITE, PROT_NONE)
CS 370 - Operating Systems – Spr	flags : Updates to the mapping should be visible to other processes mapping the same region. (MAP_SHARED, MAP_PRIVATE etc.)

shm_unlink()

Syntax: int shm_unlink(const char *name);

Arguments: name: is the memory object name to be unlinked

Example: int error = shm_unlink(shm_Name);

sprintf()

Syntax: int sprintf(char * buffer, const char * string, ...);

Arguments: string is stored in buffer

Example: sprintf(buffer, "Sum = %d", sum);

Makefile

- Following change is needed in your Makefile from Assignment 2
 - Add -lrt during compilation to call shm_open() and shm_unlink() (see point 4 from Notes in last page of Assignment 3)

Other Requirements

- Code should compile and run on CS Department computers.
- Submit all .c, along with Makefile and README.txt

Resources • <u>Read & Write with Pipe</u> <u>POSIX Shared Memory</u>

The order of print statements can be varied	<pre>Starter][178922]: contents read from the read end pipe: 10 [Starter][178922]: Created Shared memory "SHM_Lucas" with FD: 3 [Starter][178922]: Created Shared memory "SHM_Harmonic" with FD: 4 [Starter][178922]: Created Shared memory "SHM_Harmonic" with FD: 5</pre> [Lucas][178922]: The first 7 numbers of the Lucas series are: 2 1 3 4 7 11 18 [Lucas][178922]: The nth number in the lucas series is : 18 [Lucas][178922]: The sum of the first 7 numbers of the Lucas series is : 46 [HarmonicSeries][178922]: The first 7 numbers of the harmonic series are : [HexagonalSeries][178922]: The first 7 numbers of the hexagonal series are : [HarmonicSeries][178922]: The first 7 numbers of the hexagonal series are : [HexagonalSeries][178922]: The 7th value in the harmonic series is : 0.142857 [HexagonalSeries][178922]: The 7th number in the hexagonal series is : 91 [HarmonicSeries][178922]: The sum of the first 7 numbers of the hexagonal series is : 91 [HarmonicSeries][178922]: The 7th number in the harmonic series is : 0.142857 [HexagonalSeries][178922]: The sum of the first 7 numbers of the harmonic series is : 2.592857 [HexagonalSeries][178925]: The sum of the first 7 numbers of the harmonic series is : 2.592857 [HexagonalSeries][178925]: The sum of first 7 hexagonal numbers is : 252
can be varied	<pre>1 6 15 28 45 66 91 [HarmonicSeries][1789224]:The 7th value in the harmonic series is : 0.142857 [HexagonalSeries][1789225]:The 7th number in the hexagonal series is : 91 [HarmonicSeries][1789224]:The sum of the first 7 numbers of the harmonic series is : 2.592857 [HexagonalSeries][1789225]:The sum of first 7 hexagonal numbers is : 252 [Starter][1789221] : Lucas last number: 46 [Starter][1789221] : HarmonicSeries last number: 2 [Starter][1789221] : HexagonalSeries last number: 91</pre>



Acknowledgements

 These slides are based on contributions of current and past CS370 instructors and TAs, including J. Applin, L. Mendis, M. Warushavithana, S. R. Chowdhury, A. Yeluri, K. Bruhwiler, Y. K. Malaiya and S. Pallickara.