CS370 Assignment 3 Help Session

Fall 2018

-Abhishek Yeluri and Rejina Basnet

Assignment Review

- Objective
  - fork(),execl(),wait()
  - pipe()
  - shmget()
  - shmat()
  - shmctl()
- Two executables
  - Coordinator
  - Checker

Assignment Review

- Checker
  - Three command line arguments
    - Divisor
    - Dividend
    - Pipe File Descriptor
  - Checks whether the dividend is evenly divisible by the divisor.
  - Store result in shared memory buffer.
    - No WEXITSTATUS()!
- Implement Checker by itself first.
- Then implement Coordinator to manage process execution

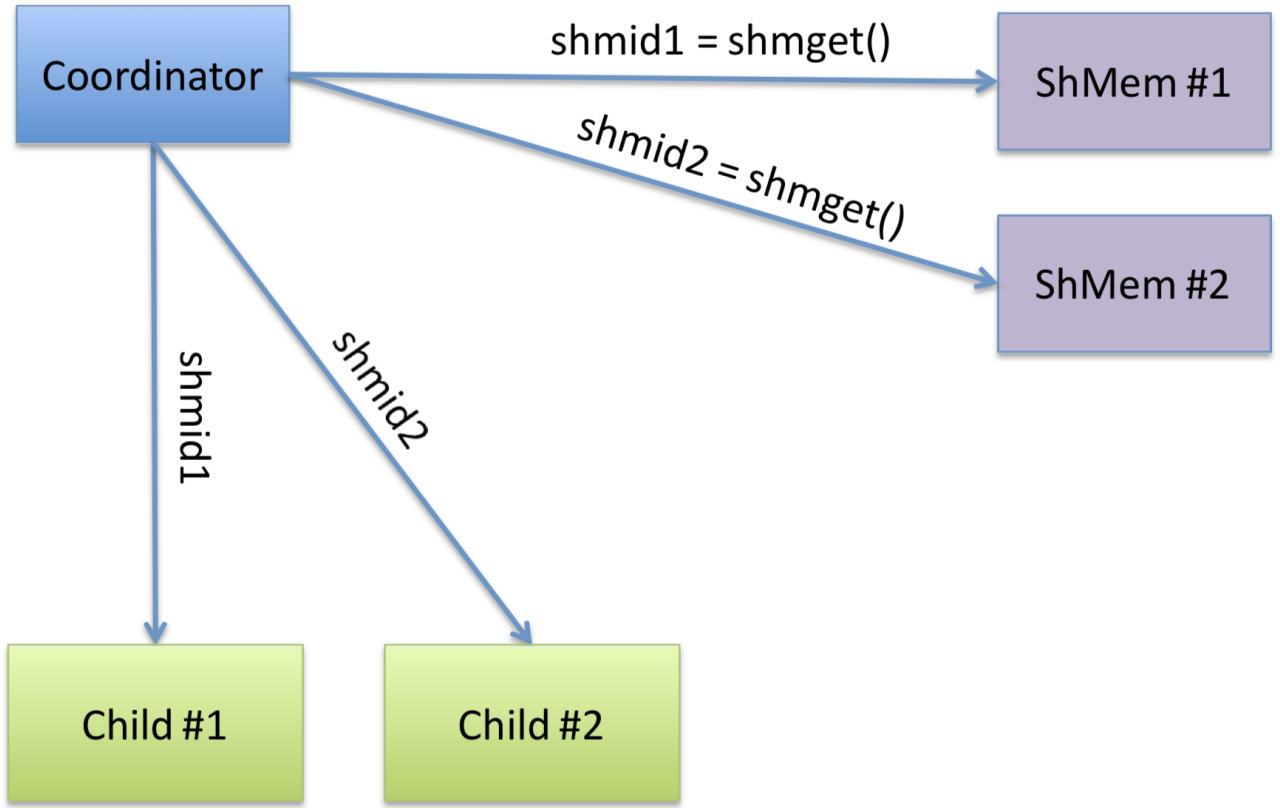
Assignment Review

- Coordinator
  - Five command line arguments
    - Divisor
    - 4 Dividends
  - 4 Child processes
    - Create shared memory segments for each child
    - Create pipe for each child process
    - fork() all 4 child processes
    - Only wait() after launching all processes!

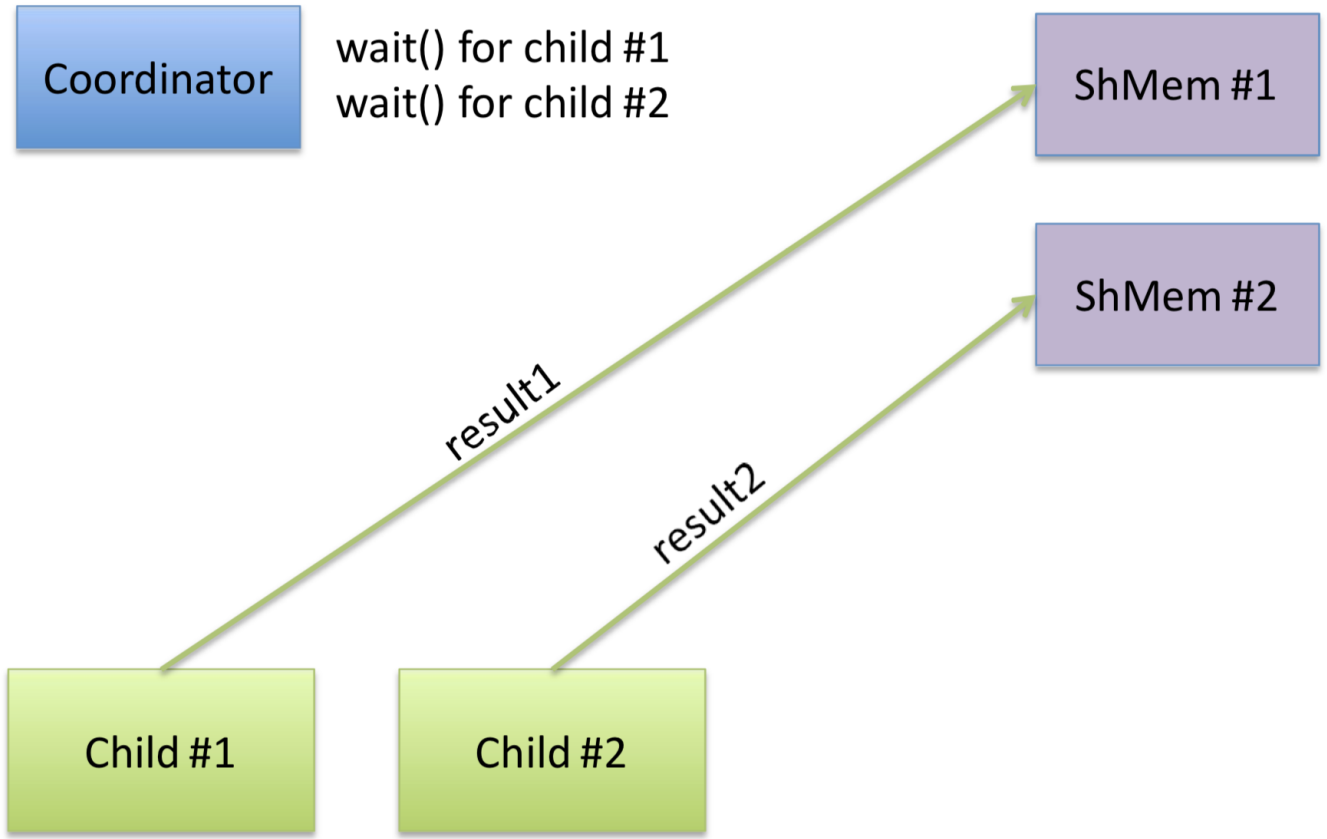
Coordinator Behavior

- pipe() – create unique pipe for each child process.
- shmget() – create shared memory segment for each child process.
- fork() all 4 child processes.
  - Send shared mem segment id through pipes.
  - Then wait().
- shmat() – retrieve results from each shared memory segment.
- shmctl() – mark shared memory to be destroyed.

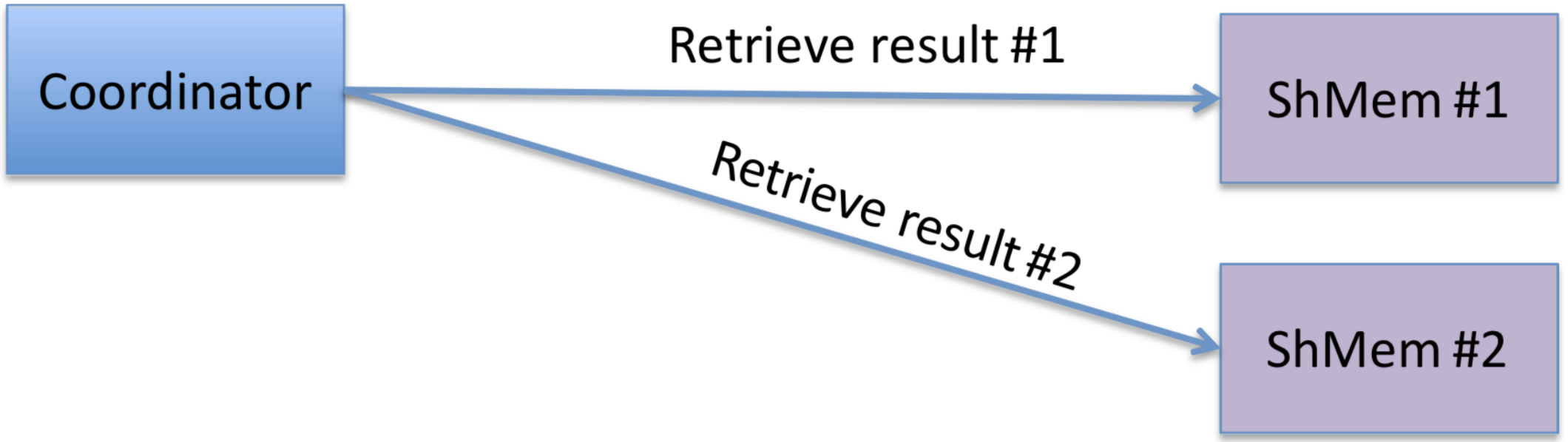
Full Picture



Full Picture



Full Picture



execl()

- Checker.c must be compiled into its own executable, e.g. “checker”.
- Use relative paths
  – If all files are in the same directory:
- execl(”./Checker", ”Checker”, argv[1], argv[i + 2], fdstr, NULL);

shmget()

int shmget(key_t key, size_t size, int shmflg);

- For this program (and most of the rest of the time), key must be IPC_PRIVATE. This is a defined constant in one of the header files. It tells the OS that it will be creating a new shared memory segment.
- Size should be big enough to hold what you're going to be storing in it (in our case, sizeof(int)).
- shmflg is composed of the following:
  - IPC_CREAT – this flag indicates that the OS needs to allocate a new chunk of shared memory.
  - “mode flags” - these are passed via a 9-bit string and are done just like binary options to chmod.

  Ex: 0664 represents 0 110 110 100.The groups of three bits represent read, write, and execute permissions for the owner, group, and world, respectively.

shmdt() and shmctl()

int shmdt(const void **shmaddr*);

- Detaches the shared memory segment located at the address specified by *shmaddr*.

int shmctl(int shmid, int cmd, struct shmid_ds *buf);

- Performs the control operation specified by *cmd* on the shared memory segment whose identifier is given in *shmid*.
- To mark the memory segment to be destroy use the command IPC_RMID.

Things to know

- The first thing you should do after fork() is close one end of the pipe in each process; in our case that means the read-end of the pipe in the parent process and the write-end of the pipe in the child process. Strange things may happen if you don't
- Once you've done shmget(), use shmat() to attach to the shared memory. Do this in both the Controller and in the Checker.
- Make sure your buffers are big enough to write to and read from the pipe!This will (obviously) affect your Checker's ability to access the shared memory.
- Clearly, it is necessary for the Controller to clear out the shared memory after each Checker instance exits.
- As a matter of good programming practice, make sure you use shmdt() and shmctl() to detach and remove the shared memory at the end of the Checker and Controller, respectively.

Requirements

- Code must run on lab machines.
- Submit all .c and .h files, along with makefile.
- MakeFile should execute make all and make clean targets.
- Use relative paths .