CS370 : OPERATING SYSTEMS

HW-2 Help Session
Assignment Review

- Objective:
  - fork()
  - exec()
  - wait()

- Two Executables:
  - Coordinator
  - Checker
Assignment Review

- Checker
  - Two command-line arguments
    1. Divisor
    2. Dividend
  - Checks whether the dividend is divisible by the divisor
  - Return 1 or 0
- Implement Checker by itself first.
- Then implement Coordinator to manage process execution.
Assignment Review

- Coordinator
  - Five command line arguments
    1. Divisor
    2. 4 Dividends
  - 4 Cycles
    1. fork()
    2. exec()
    3. wait()
Coordinator Behaviour
fork()

- Spawns a new child process
- Child copy is exact copy of parent
- Returns a different value in parent and child.
  - Child’s process id in parent
  - Zero in child
- Child can get its own process id by using getpid()
  - Must include <unistd>
exec()

- Replaces the contents of a process with a new program.
- Arguments:
  - Path and executable name
  - Executable name
  - Optional Arguments
  - Terminated by **NULL**
exec()

- execl(), execlp(), execle()
- More info: type man 2 exec in a shell
- The process is completely replaced by the new program, and the new program starts executing at its main function.
- The process ID does not change.
**wait()**

- Coordinator should wait until the child completes it's execution process.
- One checker process should be active at a time
- `wait()`
  - pid `wait(int *wstatus)`
  - `WEXITSTATUS()` allows parent to get exit status code of child (see man 2 wait)
Q: What would happen if Coordinator did not call wait()?
#CC = clang
CFLAGS += -Wall

all: checker coordinator

checker: checker.c
   $(CC) $(CFLAGS) $< -o$@

coordinator: coordinator.c
   $(CC) $(CFLAGS) $< -o$@

clean:
   rm -f checker coordinator
Requirements

- Code must run on lab machines
- Submit all .c and .h files, along with makefile
- Makefile should perform make all and make clean targets
References