COMP 3230 Midterm Q3

Common Problems

Overall Common Problems

- You cannot use sleep or while to guarantee the order
  - While causes serious performance degradation!
  - Sleep cannot guarantee the order!

- Don’t worry if you do not understand the follows, it means that you do not make these kinds of mistakes.

- `getpid()-1` may not be the parent process id

- Some student use `status = getpid()` at block A to let the child know the pid of parent
  - 1. I do not think use `status` to store a pid is a good programming habit
  - 2. You may make the same mistake as the ”Common problem 2” in slide 6

- If parent does not wait child before exiting, child may become orphan process. 2 marks will be deducted
- If you get <= 3 marks for the code part. It means that you do not use the signal correctly. The marks are given for the shared memory usage. You should learn more about multiprocessing programming!

Common Problem 1

- Pattern:
  - Child stops itself in each iter
  - Parent does NOT wait child but just send SIGCONT to child in each iter
- Problem:
  - Before child stops itself, the parents can finish its loop execution
  - See the figure in the next slide

Code segment	C code to be added by you (or lea
A	// initialization shmid = shmget(IPC_PRIVATE, 10

B	// child:
	
	b = shmat(shmid, 0, 0);

C	// child: for-loop
	
	raise(SIGSTOP);

D	// child: after for-loop
	
	shmdt(b);
	
	exit(0);

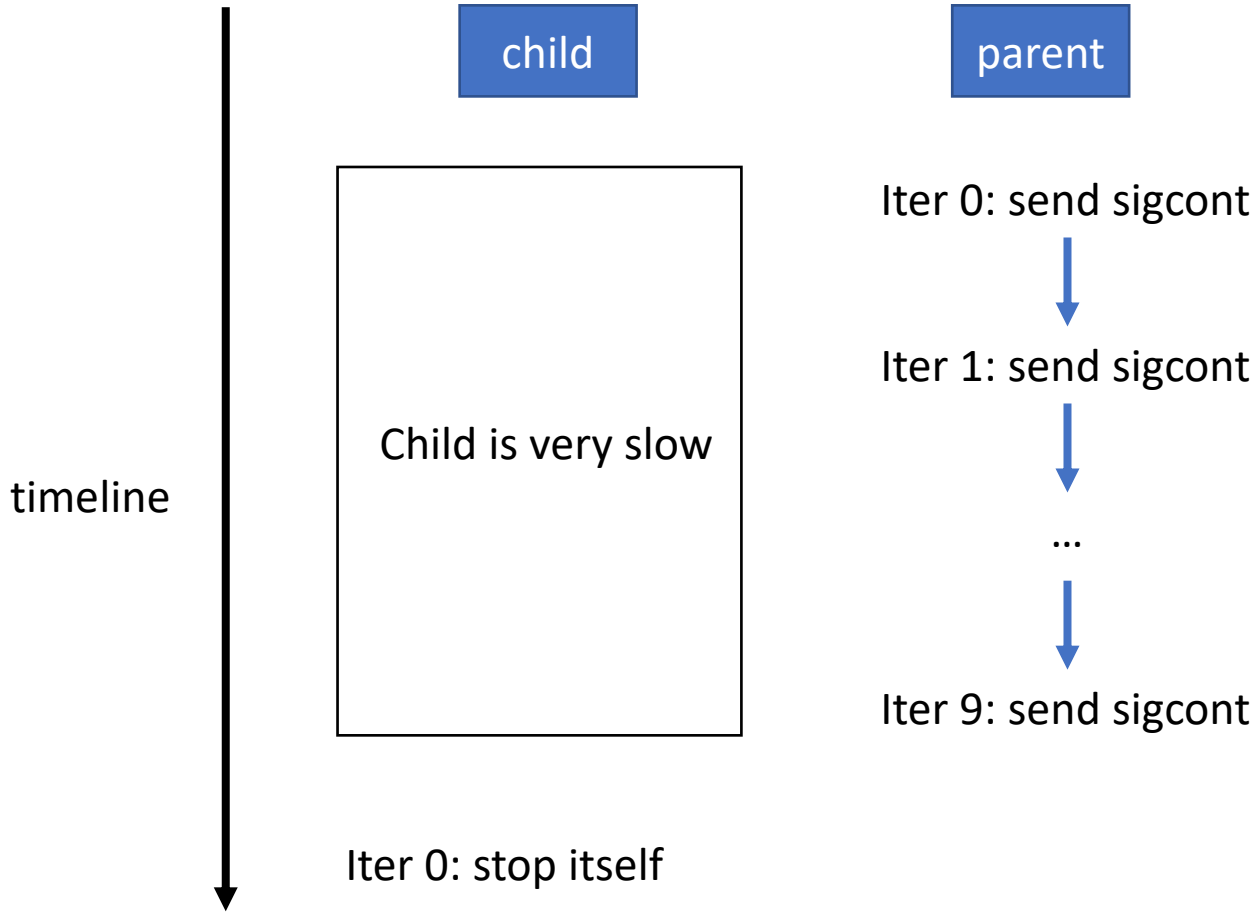
E	// parent
	
	a = shmat(shmid, 0, 0);

F	// parent: for-loop (before a[i]= i)

G	// parent: for-loop (after printf())
	
	kill(pid, SIGCONT);

H	// before the end of the program
	
	shmdt(a);
	
	shmctl(shmid, IPC_RMID, NULL);

Common Problem 1



Common Problem 2

- Pattern:
  - Child continues parent and then stops itself in each iter
  - Parent continues child and then stops itself in each iter
- Problem:
  - getppid() is not allowed according to the specs
  - **<u>And it is not correct!</u>**
  - I only give comments like “should not use getppid” in your submission, and the main reason is given in the next slide (too long to be put in comment)

B	// child

C	// child: for-loop
	`/* C: Anything to be done here? */`
	`kill(getppid(), SIGCONT);`
	`kill(getpid(), SIGSTOP);`

D	// child: after for-loop
	`/* D: Anything to be done by Child after the fo`
	`kill(getppid(), SIGCONT);`
	`shmdt(a);`
	`shmdt(b);`
	`exit(EXIT_SUCCESS);`

E	// parent
F	// parent: for-loop (before a[i]= i)

G	// parent: for-loop (after printf())
	`/* G: Anything to be done here? */`
	`kill(pid, SIGCONT);`
	`kill(getpid(), SIGSTOP);`

H	// before the end of the program
	`/* H: Anything to be done at the end of the prog`
	`wait(NULL);`
	`shmdt(a);`
	`shmdt(b);`
	`shmctl(shmid, IPC_RMID, NULL);`
	`exit(EXIT_SUCCESS);`

Common Problem 2