

Quiz #4

Collaboration with classmates is allowed. Upload your program with you name and ID at the top of the file.

The program “demo.c” is provided on Canvas and demonstrates how to start “top” using `execvp()`. You will write a program where a parent and child process do work concurrently while they are monitored using “top”. Write a single program that does the following:

1. The parent creates a child process that runs “top” using `execvp()`. This will monitor all processes.
2. The parent creates another child process that increments a counter by 1 for a total of N_C times and stores the final value in shared memory. Choose any value for N_C that you wish. The final value of the counter will be N_C , but it should take a few seconds to finish such that the process appears in “top”.
3. The parent increments a counter by 1 for a total of N_P times. Choose any value for N_P that you wish. The final value of the counter will be N_P , but it should take a few seconds to finish, but it should take a few seconds to finish such that the process appears in “top”.
4. When both children have terminated (use “wait(NULL)” for both children), the parent prints the sum of $N_C + N_P$. Note that “top” will not terminate until the user types “q”.

When the program is first executed, it might initially show something like the following, where the user “abc12” named the program “quiz4.c”:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
11011	abc12	20	0	8468	768	668	R	99.4	0.0	0:00.20	quiz4
11012	abc12	20	0	8468	116	0	R	98.7	0.0	0:00.16	quiz4
11013	abc12	20	0	20256	2648	2156	R	5.4	0.1	0:00.02	top
6692	abc12	20	0	103828	4572	3672	S	0.0	0.2	0:00.00	sshd

After the loops in steps (2) and (3) have finished, “top” would show the following while the parent waits for the termination of the process that is still running “top”:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
11011	abc12	20	0	8468	768	668	R	0.0	0.0	0:00.20	quiz4
11013	abc12	20	0	20256	2648	2156	R	5.4	0.1	0:00.02	top
6692	abc12	20	0	103828	4572	3672	S	0.0	0.2	0:00.00	sshd

Supposing that $N_C = 1000000000$ and $N_P = 1000000000$, the user types “q” and the following should appear immediately afterwards:

sum = 2000000000

Note: Because use must use shared memory, remember to use “-lrt” when compiling.