Real Time Systems, January 2024

Synchronizing termination of processes

Opened: Monday, 22 January 2024, 9:00 AM
Due: Friday, 2 February 2024, 11:59 PM

✔ Done

In the previous assignment you have written a C program which when executed as "***./a.out executable1 executable2 ... executableN***" creates **N** additional child processes where the 1st child process executes the 1st executable file (given by ***executable1***). 2nd executes the 2nd executable file (given by ***executable2***) and so on.

In this assignment modify the program so that the parent process waits for completion of its **N** child processes and prints the status with which the child processes terminated along with the process id of the corresponding child process. **Observe and report** the status reported by a child process that terminates normally and one that terminates abnormally. Nnote that the first (least signficant) 8 bits of the status "reported" by a terminated child process are used by the OS (kernel) to signify abnormal termination!

Submission status

Attempt number	This is attempt 1.
Submission status	Submitted for grading
Grading status	Not graded
Time remaining	Assignment was submitted 2 days 22 hours early
Last modified	Wednesday, 31 January 2024, 1:21 AM
Online text	✚ (40 words) Abnormally terminated processes send NonZero status to the parent process. Normally terminated processes return Zero status to the parent ...
File submissions	program7.c 31 January 2024, 1:21 AM
Submission comments	▸ Comments (0)

◄ Child Processes execute Programs with command line arguments (say, with switches)

Jump to...

Synchronizing termination of processes and using the status reported by a child process at the time of termination ►