

CS 307 System Practicum
ASSIGNMENT 3
(Process-1)

NOTE : Avoid using direct system calls to implement the commands.

1. Create four jobs, one job to edit a file, two CPU hogs and one sleep process.
[a]. Set the jobs to different states and observe the corresponding process status.
[b]. Try to edit a file when the job is in different states
[c]. Observe CPU utilization of the processes
[d]. Create four more CPU hogging process and observe CPU utilization
[e]. Change priorities of processes and observe CPU utilization and priorities
2. Write a program (parent) that creates a child process and sleeps for x sec. The child sleeps for y sec. Adjust values of x and y to answer following questions :
[a]. What is the process status of child if it terminates before its parent?
[b]. When parent terminates before child, what is PPID (Parent Process Id) of the child?
[c]. What function is used to make parent wait for child to terminate?
[d]. Can you rewrite the parent using exec? What are differences between fork and exec?
3. Write a program to create two children, four grandchildren, eight great grandchildren. Every process except for great grand children will create exactly two children. The parent takes a positive integer (maxInt) as command line argument. The parent takes a positive integer (maxInt) as command line argument. Each children will print all prime number between 0 to maxInt, similarly each grandchildren will print Fibonacci series up-to maxInt and great grand children will print the intersection of above two in an interval of 1 second. Output of each process should be easily distinguishable. What will happen if you kill a grandchild, a child.
Note: maxInt is inclusive in each case.
4. Create four children C1 to C4, each having a pipe to communicate with the parent, where :

[i] C1 reads from keyboard and encrypt it using a mapping table and send it to Parent (eg A converted to F, G to I etc). It reads the mapping table from a predefined file.

[ii] C2 reads data from a text file and speaks it using "espeak" linux utility.

[iii] C3 copies a given input file. Here do not use the linux "cp" utility. Code the copy function.

[iv] C4 does CPU monitoring (total %CPU used etc) and periodically sends stats to the parent.

The parent reads from the four pipes and prints on the display. (Hint: use select())

Optional : Write a script to monitor CPU utilization of the above program.