

HOMEWORK 1

U0942570-Vairavan Sivaraman

2.13 Describe three general methods for passing parameters to the operating system.

1. Using registers, we can directly pass parameters to the operating systems.
2. With the help of stack, we can push the parameters into the stack(Basically store the parameters into stack) and pop it using the operating systems.
3. We can store all the parameters in a table and then table address will be stored in register. Then OS will access the parameters with the help of memory address in the register.

2.14 Describe how you could obtain a statistical profile of the amount of time spent by a program executing different sections of its code. Discuss the importance of obtaining such a statistical profile.

1. Operating systems comes with a timing profile to estimate the amount of time spent at each part of program.
2. A timer profile uses a tracing facility or timer interrupts provided by the System.
3. The counter is recorded for every time interrupts. With enough number of timer interrupts, we can obtain the statistical picture of time spent by the program at various parts.

3.9 Describe the actions taken by a kernel to context-switch between processes.

The actions taken by a kernel to context-switch between processes are:

- Save the context of current process context running on the CPU.
- PCB represents context of process. All fields in PCB are saved to restore when the process returns back to running state.
- Load the saved context (PCB) of the process, which is scheduled to run.

3.10 Construct a process tree similar to Figure 3.8. To obtain process information for the UNIX or Linux system, use the command `ps -ael`. Use the command `man ps` to get more information about the `ps` command. The task manager on Windows systems does not provide the parent process ID, but the process monitor tool, available from technet.microsoft.com, provides a process-tree tool.

1. Stored the `ps -ael` output into a file.
2. Using Awk program, extracted the PID, PPID,CMD fields into an another file.
3. Using C program constructed the .DOT file.
4. Using the DOT terminal command built a .ps file.
5. The image is attached in a separate file. (output.ps)

3.17 Using the program shown in Figure 3.35, explain what the output will be at lines X and Y .

Line X output is mentioned below:

CHILD: 0 CHILD: -1 CHILD: -4 CHILD: -9 CHILD: -16

In Line X, the child process segment is invoked and the child process copies the array into to the data segment of child process. The child process changes the values of array by multiplying the array elements with itself and a negative sign

Line Y output is mentioned below:

PARENT: 0 PARENT: 1 PARENT: 2 PARENT: 3 PARENT: 4

In Line Y, the parent process waits for the child to complete and prints the array elements. Although child process changed the values of array, Child process made a copy of all the data and changed its own array. So, the child process cannot change the array of parent. Hence it prints 0,1,2,3.

3.21

The program is attached along with this submission on Canvas.The file is named as collatz.c