**Rectification**

**Updated assignment 1:**

MultiProcessing

1. Where are the function arguments and variables stored?

function arguments and variables sare stored **on the stack,** they pushed immediately before the return address. Everything what lives on the stack (local variables, parameters etc.)

1. Where are global variables stored?

Global variables are stored **in the data section.** Unlike the stack, the data region does not grow or shrink — storage space for globals persists for the entire run of the program. Finally, the heap portion of memory is the part of a program's address space associated with dynamic memory allocation.

1. What are the resources assigned to a process?

The process needs certain resources such as **CPU and memory** to perform the tasks.

1. How are processes identified?

They are identified using process id or PID

1. Who selects the process for execution?

**CPU scheduler** selects a process among the processes that are ready to execute and allocates CPU to one of them.

1. What are the guiding principles used by scheduler to select a process?

Fairness − All processes should be treated the same. No process should suffer indefinite postponement. Maximize throughput − Attain maximum throughput**.**

1. List atleast 5 scheduling algorithms.

Six types of process scheduling algorithms are: First Come First Serve (FCFS), 2) Shortest-Job-First (SJF) Scheduling, 3) Shortest Remaining Time, 4) Priority Scheduling, 5) Round Robin Scheduling, 6) Multilevel Queue Scheduling.

1. What do you mean by single and multi core?

A single-core processor is a microprocessor with a single core on its die. It performs the fetch-decode-execute cycle once per clock-cycle, as it only runs on one thread. A computer using a single core CPU is generally slower than a multi-core system.

A processor that has more than one core is called Multicore Processor while one with single core is called Unicore Processor or Uniprocessor. Nowadays, most of systems have four cores (Quad-core) or eight cores (Octa-core)

1. How many processes can a N core CPU run parallely?

You must have more than one processing core to execute **two processes** in parallel. Erlang is built for concurrency and will run concurrent solutions (even with a single CPU). Given multiple execution cores, it can also execute processes in parallel.

1. How is a program executed internally? What are the steps involved?

1)C program (source code) is sent to preprocessor first. The preprocessor is responsible to convert preprocessor directives into their respective values. The preprocessor generates an expanded source code.

2) Expanded source code is sent to compiler which compiles the code and converts it into assembly code.

3) The assembly code is sent to assembler which assembles the code and converts it into object code. Now a simple.obj file is generated.

4) The object code is sent to linker which links it to the library such as header files. Then it is converted into executable code. A simple.exe file is generated.

5) The executable code is sent to loader which loads it into memory and then it is executed. After execution, output is sent to console.

1. What are the various attributes of a process? Mention atleast one command to view process attributes

1.Process ID · 2. Program counter · 3. Process State · 4. Priority · 5. General Purpose Registers · 6. List of open files · 7. List of open files 8. List of open devices

Command - ls

1. What are the different states of a process?

Running − In this state the instructions are being executed. Waiting − The process is in waiting state until an event occurs like I/O operation completion or receiving a signal. Ready − The process is waiting to be assigned to a processor. Terminated − the process has finished execution.

1. How do we run multiple processes using a single CPU?

Single CPU systems **use scheduling and can achieve multi-tasking** because the time of the processor is time-shared by several processes so allowing each process to advance in parallel. So a process runs for some time and another waiting gets a turn.

1. What do you mean context switch? When does it happen?

A context switching is **a process that involves switching of the CPU from one process or task to another**. In this phenomenon, the execution of the process that is present in the running state is suspended by the kernel and another process that is present in the ready state is executed by the CPU.

1. What does the term concurrency and parallelism mean?

Concurrency is the task of running and managing the multiple computations at the same time. While parallelism is the task of running multiple computations simultaneously.

1. Why do we need to assign priorities to processes?

Establishing priorities is necessary **in order to complete everything that needs to be done**. Prioritization is important because it with allow you to give your attention to tasks that are important and urgent so that you can later focus on lower priority tasks.

1. Which command is used to view process status in realtime?

**ps command** (ps means process status).

1. Which command is used to view process tree with pid details?

The **tasklist command**

Use the built in Windows tasklist command from a command prompt to display all processes, their PIDs, and a variety of other details

1. Which command is used to get pid, ppid and process group id?

**Ps -la command** is used to list the currently running processes and their PIDs along with some other information depends on different option

1. Which process starts all processes in the system?

**Whenever a command is issued in Unix/Linux, it creates/starts a new process**. For example, pwd when issued which is used to list the current directory location the user is in, a process starts. Through a 5 digit ID number Unix/Linux keeps an account of the processes, this number is call process ID or PID.

1. How to create a new process from within a program?

A new process can be created by the **fork() system call**. The new process consists of a copy of the address space of the original process. fork() creates new process from existing process. Existing process is called the parent process and the process is created newly is called child process.

1. Where the process information maintained? What is the name of the data structure used to hold process information?

All processes are stored **in the job queue**. Processes in the Ready state are placed in the ready queue.  The **symbol table** is a data structure that is used to hold information about source code during the compilation process.

1. What happens on exit()?

On many computer operating systems, **a computer process terminates its execution** by making an exit system call. More generally, an exit in a multithreading environment means that a thread of execution has stopped running. For resource management, the operating system reclaims resources (memory, files, etc.)

1. What is the difference between exit() and \_exit()? Which will cause quick exit?

The \_Exit() function in C/C++ gives normal termination of a program without performing any cleanup tasks.

**exit() function performs some cleaning before termination of the program like connection termination, buffer flushes etc**

1. Does \_exit close open fds?

\_exit() **closes all open file descriptors and directory streams in the caller**.

1. Does \_exit flush open streams?

**No, exit should not flush iostreams**. iostreams are flushed on close() (on the stream types where it is available), when flush is called explicitly on the stream, or on destruction.

1. What happens when you press Ctrl+C?

It will Kill the process

1. What happens when you press Ctrl+Z?

Ctrl+Z in Word and other word processors.In Microsoft Word and other word processors, pressing Ctrl + Z will **undo any change made in a document**.

1. What is the use of an fd? How is it different from FILE \*?

You pass "naked" file descriptors to actual Unix calls, such as read() , write() and so on. **A FILE pointer is a C standard library-level construct, used to represent a file**. The FILE wraps the file descriptor, and adds buffering and other features to make I/O easier.

1. How many fd’s are created for every process? What are they?

After the directory server has exceeded the file descriptor limit of **1024 per process**, any new process and worker threads will be blocked.

1. Name the call to get an fd for a file

fdopen manual . Get the file descriptor from a FILE pointer (e.g. file ) in C on Linux: **int fd = fileno(file);** More details can be found in the man page of fileno : fileno manual .

1. If a process creates a child sub process, how can it detect exit of a child?

You can get the exit status of the child **via the first argument of wait() , or the second argument of waitpid() , and then using the macros WIFEXITED and WEXITSTATUS with it**. waitpid() will block until the process with the supplied process ID exits.

1. Which process reaps the exit code of orphan child?

The **parent process** reads the exit status of the child process which reaps off the child process entry from the process table. Orphan Process: A process whose parent process no more exists i.e. either finished or terminated without waiting for its child process to terminate is called an orphan process.

1. What all does a child inherit from its parent?

A child process inherits **the current directory of its parent process by default**. However, CreateProcess enables the parent process to specify a different current directory for the child process. To change the current directory of the calling process, use the SetCurrentDirectory function.