**Threads**

&used to run at background

LWT threads

LWT process

Concureency – to ability of system and performing multiple tasks

Concurrency

Responsiveness, Utilizing Multi-core Processors (web server) ,utilizing multi-core processors (distributing tasks),efficient resource utilization(how we use the core processors) , scalability (efficiently or effectively)

….what is concurrency in programming…..

Multithreading : different task perform in same process parallel , ahre the **sam memory space which allow same allocation** but care synchronization to **prevent race conditions**…. (deposit withdraw wxample);(upredictable value)

Multiprocessing : from one process we create multiple process(1-preoce -addapp(in one proc) and 2nd sub(in ome proce) (bothe are indepent process ) …. The proce have there **own memory spaces(ipc inter process communication)(it is a shared memory between the process)(it can be share by pipes or shared memory )**

Asynchronous Programming : it doesn’t depend on the other

Concurrency vs parallelism

1. Concurrency is dealed with multiple tasks once or having multiple tasks in progress
2. Parralesim is about executing multiple tasks truly(have different memory allocation or multi process),typically is achieved **by multiple cpu cores**

**How we can achieve ….**

Multi threading – using POSIX threads

Multi processing : use fork() – creates separate process with individual memory spaces

Asynchronus i/o: use unbloxking i/o operations for concurrent processing

Atomic operations: enables simple thread – safe operations without full mutex overloade

………………

Concept of threading :

Samllest threads in treading it will have its own **stack , register ,program counter**

Threads can easily communicate and ahre the data because have same memory

**Process = process context + core+data+stack**

**Stack**

**Shared lib**

**break**

**Runtime**

**Read / write**

**Pc 🡪Read only**

Thread (main thread )

Data registers

Condition codes

Sp1

Pc1

Each thread has its own id

Thread vs Processes

Similarities

1. Each has its own logic
2. Can run concurrently
3. Context switching

Differences

1. Threads share code, data , process do not
2. Threads some what less expensive
3. Process is twicing expensive as thread

Thread process

Shares same memory has its own memory space

Light weight and faster slower to crate and manage

**Differentnt method used posix ( Pthreads)**

1. **Creating and reaping threads**
2. Pthread\_create, pthread\_join(wait or attach)
3. **Determining your thread id**
4. Pthread\_self
5. **Terminating threads**
6. Pthread\_cancel, pthread\_exit
7. Exit(terminates all treads ),return (terminates current thread)
8. **Synchronizing access to shared variabkes**
9. Pthread\_mutex\_init, pthread\_mutex\_[un]lock
10. Pthread\_con\_init,pthread\_cond\_[timed]wait

…..we have to use pthread.h

On error --- error (insufficient resources to create)

If it success it reaturns 0

Pthread\_create(&tid , NULL , howdy , NULL) --- (here i

A screen shot of a computer code

Description automatically generated

A computer screen shot of a program code

Description automatically generated

1. Structure iwas to type casted to void and it should be passed to employee as base struct and it should be call by base address

RASE Condition is occurs only when there is

LINUX COMMAND LINES……..

Main(int argc,char\*argv[],char \*enc[])//(int argc,char\*argv[])

A computer screen shot of a program code

Description automatically generated