ORANE LABS

SHC HT KANPUR

www.oranelabs.com

ORANE LABS

SHC HT KANPUR

ORANE LABS

SHC HT KANPUR

### ORTHREAD SYNCHRONIZATION AND LABS

SHC HT KANPUR

SHC HT KANPUR

ORANE LABS ORANE LABS

ORANE LABS

www.oranelabs.com

www.oranelabs.com

### ORANE LABS ORANE LABS

SHC HT KANPUR

SHC HT KANPUR

SHC HT KANPUR

- www.When multiple threads of control share the same com memory, we need to make sure that each thread sees a consistent view of its data.
- Consistency problem arises when one thread KANPUR www.modifies a variable that is being shared by anelabs.com another thread. ORANE LABS
- If each thread uses variables that other threads don't read or modify, no consistency problems ANPUR www.wilhexist.com

ORANE LABS

ORANE LABS ORANE LABS

Interleaved memory cycles with two orthreads\BS ORANE LABS ORANE LABS SHC HT KANPUR www.oranelabs.com ORANE L Thread A Thread B DRANE LABS SHC HT KA read DRANE LABS ORANE L write, SHC HT KA read write, ORANE LADS ORANE LABS

www.oranelabs.com www.oranelabs.com www.oranelabs.com

# Locks to the rescue ORANE LABS ORANE LABS

- To solve this problem, the threads have to use UR a lock that will allow only one thread to access the variable at a time.
  - If it wants to read the variable, thread B acquires a lock.
    - Similarly, when thread A updates the
- thread B will be unable to read the variable until thread A releases the lock.

ORANE LABS ORANE LABS ORANE LABS

Two threads synchronizing memory access ORANE ORANE LABS SHC HT KANPUR Thread A Thread B read ORANE ORANE LABS SHC HT KANPUR SHC HT read write, time ORANI ORANE LABS SHC HT write, read

www.oranelabs.com www.oranelabs.com

ORANE LABS

# Synchronization with Semaphores

- A semaphore is a variable that is used for HT KANPUR www.controllingaccess, by multiple processes, to alabs.com common resource in a parallel programming or a multi user environment.
  - - There are two types of semaphore:

ORANE

- Binary Semaphore: takes value of 0 or 1.
- ORA Counting Semaphore: Wider range of values. LABS

SHC HT KANPUR SHC HT KANPUR

For synchronizing we will be using binary semaphores.

ORANE LABS ORANE LABS ORANE LABS

www.oranelabs.com www.oranelabs.com Semaphore Functions The semaphore functions do not start with ANPUR www.pthread\_c, as most thread-specific functions abs.com OR do, but with seme ANE LABS ORANE LABS SI A semaphore is created with the sem\_init KANPUR function which is declared as follows: oranelabs.com WWW. #include < semaphore.h> int sem\_init(sem\_t \*sem, int pshared, unsigned int value); RANE LABS www.oranelabs.com ORANE LABS ORANE LABS ORANE LABS www.oranelabs.com www.oranelabs.com Next Set of functions ORANE LABS SI • The next pair of functions controls the value of the T KANPUR semaphore and is declared as follows: #include < semaphore.h> LAB5 intsem wait(sem t \* sem); intsem post(sem t \* sem); rnese both take a pointer to the semaphore object initialized by a call to sem\_init. The sem post function atomically increases the value of the semaphore by 1. RANE LABS ORANE LABS The sem wait function atomically decreases the value of SHO the semaphore by one, but always waits until the HT KANPUR semaphore has a nonzero count first. ORANE LABS ORANE LABS ORANE LABS

www.oranelabs.com

### ORANE LABS

ORANE LABS

Suppose we are creating an application in KANPUR www.which one thread accepts input from the user com or and another thread counts the number of LABS characters entered. HT KANPUR SHC HT KANPUR

 Now in such a scenario, the two threads will be accessing a common resource, the

character array:

- char work\_area[WORK\_SIZE]

Now since one thread is modifying this array

and another thread is reading at the same

www.oranelabs.com

### ORANE LABS

ORANE LABS

SHC HT KANPUR

sem\_t bin\_sem; //A semaphore

www.oranelabs.com

• Initializing the semaphore: <sup>B5</sup>

ORANE LABS

SIIC IIT res = semRinit(&bin sem, 0, 0); PUR

 $www.oranif(res.!=0){$ 

perror("Semaphore initialization failed");

exit(EXIT\_FAILURE);

ORANE LABS

SIIC IIT KANPUR

ORANE LABS

ORANE LABS

Creating The thread ORANE LABS SIIO IITres = pthread\_create(&a\_thread, NULL, thread | function, PUR NULL); ww.oranelabs.com if (res!= 0) { www.oranelabs.com perror("Thread creation failed"); ORANE LABS exit(EXIT\_FAILURE); HT KANPUR www.oranvoid \*thread\_function(void \*arg) { SIIC IIT printf("You input %d characters\n", strlen(work\_area) -1); UR www.oransem\_wait(&bin\_sem);anelabs.com ORANE LABS ORANE LABS www.oranelabs.com www.oranelabs.com Code for the main Thread ORANE LABS ORANE printf("Input some text. Enter 'end' to finish\n"); T KANPUR while(strncmp("end", work\_area, 3) != 0) { www.oranelabs.com fgets(work area, WORK SIZE, stdin); ORAsem\_post(&bin\_sem); NE LABS ORANE LABS fit Kanpur SHC HT KANPUR SHC HT KANPUR printf("\nWaiting for thread to finish...\n"); www.ores = pthread\_join(a\_thread, &thread\_result); oranglabs.com if (res != 0) { perror("Thread join failed"); ORANE LABS exit(EXIT PFAILURE); IC IIT KANPUR www.oranelabs.com www.oranelabs.com printf("Thread joined\n"); ORANE LABS

www.oranelabs.com Closing the semaphore ORANE ORANE LABS SHC HT KANPUR sem\_destroy(&bin\_sem); ORANE LABS exit(EXIT\_SUCCESS); KANPUR ORANE LABS ORANE LABS ORANE LABS SHC HT KANPUR SHC HT KANPUR www.oranelabs.com ORANE LABS ORANE LABS ORANE LABS www.oranelabs.com www.oranelabs.com ORANE LABS ORANE LABS SUO When we initialize the semaphore, we set its NPUR www.value.to 0. Thus, when the thread's function become starts, the call to sem\_wait blocks and waits for the semaphore to become nonzero.

- for the semaphore to become nonzero.

  In the main thread, we wait until we have some text and then increment the semaphore with sem\_post, which immediately allows the
- other thread to return from its sem\_wait and start executing.
- Once it has counted the characters, it again
   calls sem\_wait and is blocked until the main

www.oranelabs.com Protecting Accesses to Shared Variables: Mutexes ORANE LABS ORANE LABS ORANE LABS SILO Problem Statement: IT KANPUR SHC HT KANPUR This program creates two threads, each of which com or an executes the same function. ORANE LABS — The function executes a loop that repeatedly increments a global variable, glob, by copying glob into the local variable loc, incrementing loc, and ORAN copying loc back to glob. LABS ORANE LABS SHC HT KANPUR SHC HT KANPUR www.oranelabs.com ORANE LABS ORANE LABS ORANE LABS The Program www.oranelabs.com www.oranelabs.com #include <pthread.h> ORANE LABS ORANE LABS static int glob = 0; SII static void \* UR \* Loop 'arg' times incrementing NPUR www.o'glob'1\*/s.com www.oranelabs.com www.oranelabs.com threadFunc(void \*arg) ORANE LABS int loops = \*((int \*) arg); SHC HT KANPUR www.oranelabs.com www.oranelabs.com www.oranolabs.com  $ORAfor(j = 0; j < loops; j++) \{ IE LABS \}$ ORANE LABS II loc = glob; loc++; glob = loc; ORANE LABS ORANE LABS

int main(int argc, char \*argv[]) ORANE LABS ORANE LABS ORANE LABS pthread t t1, t2; int loops =1000, s; SHC HT KANPUR s = pthread\_create(&t1, NULL, threadFunc, &loops); if (s!=0)ORANE LABS s = pthread\_create(&t2, NULL, threadFunc, &loops); SIIC IIT KANPUR SHC if (s != 0) NPUR printf( "pthread\_create");
www.oranelabs.com
s = pthread\_join(t1, NULL); if (s != 0)printf(s, "pthread\_join"); ORANE LABS if (s!=0)www.oraprintf(s, "pthread\_join"), ranelabs.com printf("glob = %d\n", glob); ORANE LABS exit(EXIT SUCCESS);

Current Thread 1 Thread 2 value of globRepeatedly: loc = glob; loc++; glob = loc; 2000 loc = glob; time slice time slice expires begins Repeatedly: loc = glob; loc++; glob = loc; 3000 time slice time slice ends begins loc++; Key glob = loc; 2001 Waiting Executing on CPU for CPU

www.oranelabs.com

www.oranelabs.com

www.oranelabs.com

## Mutexes

ORANE LABS

ORANE LABS

- We can protect our data and ensure access by UR www only one thread at a time by using the nelabs compthreads mutual-exclusion interfaces.
- A mutex is basically a lock that we set (lock)

  SIC before accessing a shared resource and release (unlock) when we're done.
- While it is set, any other thread that tries to set it will block until we release it.
- If more than one thread is blocked when we unlock the mutex, then all threads blocked on the lock will be made runnable, and the first one to run will be able to set the lock.

www.oranelabs.com

www.oranelabs.com

www.oranelabs.com

### ORANE LABS

SHC HT KANPUR

www.oranelabs.com

## The Protocol

SHC HT KANPUR

www.oranelabs.com

### ORANE LABS

SHC HT KANPUR

www.oranelabs.com

• Each thread employs the following protocol

SHC for accessing a resource: ANPUR SHC HT KANPUR

www.o-lock the mutex for the shared resource; voranelabs.com

ORA- access the shared resource; and

SIIC II KANPUR mutex. IIT KANPUR

www.oranelabs.com www.oranelabs.com

ORANE LABS ORANE LABS

ORANE LABS

SIIC IIT KANPUR

www.oranelabs.com

Protecting the Critical section Thread A Thread B www.oranelabs.com lock mutex M ORANE LABS lock mutex M blocks Saccess shared resource unlock mutex M unblocks, lock granted ORANE LABS access shared resource unlock mutex M ORANE LABS www.oranelabs.com www.oranelabs.com www.oranelabs.com Initializing the mutex ORANE LABS A mutex variable is represented by the KANPUR www.pthread mutex t data type. www.oranelabs.com ORANE LABS ORANE LABS ORANE LABS #include < pthread.h > int pthread\_mutex\_init(pthread\_mutex\_t \*restrict mutex , const pthread\_mutexattr\_t \*restrict attr); intpthread\_mutex\_destroy(pthread\_mutex\_t \* mutex); Both return: 0 if OK, error number on failure SHC HT KANPUR

ORANE LABS

ORANE LABS

www.oranelabs.com Locking and Unlocking a mutex To lock a mutex, we call PUR www.pthread mutexwlock. If the mutex vis ranelabs.com already locked, the calling thread will block as until the mutex is unlocked. To unlock a mutex, we call pthread\_mutex\_unlock. #include < pthread.h > intpthread mutex lock(pthread mutex t\* mutex); intpthread mutex trylock(pthread mutex t\* mutex); intpthread mutex unlock(pthread mutex t\* mutex); SHC All return: 0 if OK, error number on failure ORANE LABS ORANE LABS ORANE LABS www.oranelabs.com www.oranelabs.com ORANE LABS ORANE LABS ORANE LABS SHC HT KANPUR SHC HT KANPUR Inter-Process www.oranelabs.com ORANE LABS SHC HT KANPUR ORANE LABS ORANE LABS ORANE LABS

SIIC IIT KANPUR

ORANE LABS

SIIC IIT KANPUR

ORANE LABS

SIIC IIT KANPUR www.oranelabs.com

www.oranelabs.com

## What is IPC

### ORANE LABS

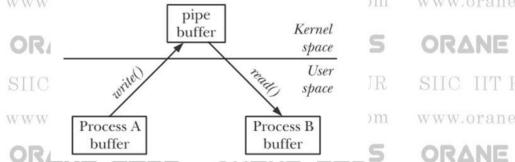
ORANE LABS

- A mechanism used by processes to HT KANPUR www.communicate.with each other.
- A Taxonomy of IPC Facilities: ORANE LABS
- SIIC + Communication: These facilities are concerned ANPUR www.orawith exchanging data between processes.
- Synchronization: These facilities are concerned with synchronizing the actions of processes or SHC HT KANPUR SIIC II'threads.UR
- www.o+Signals: Although signals are intended primarily bs.com for other purposes, they can be used as a synchronization technique in certain

www.oranelabs.com www.oranelabs.com

### Communication Facilities DRANE LABS ORANE LABS

- We can break the communication facilities ANPUR www.intotwocategories:ranelabs.com www.oranelabs.com
- ORA- Data-transfer facilities: The key factor RANE LABS distinguishing these facilities is the notion of KANPUR writing and reading



ORANE LABS

### ORANE LABS ORANE LABS ORANE LABS

SHC HT KANPUR

SHC HT KANPUR

www.spanelabs.com www.oranelabs.com www.oranelabs.com

— Shared memory allows processes to exchange SHO Hinformation by placing it in a region of memory NPUR that is shared between the processes.

- Because communication doesn't require system calls or data transfer between user memory and SHC Hikernel memory, shared memory can provide very PUR fast communication.

ORANE LABS

ORANE LABS

ORANE LABS

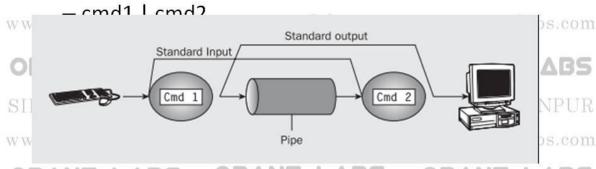
www.oranelabs.com www.oranelabs.com

www.oranelabs.com

### What Is a Pipe? ORANE LABS

ORANE LABS

We use the term pipe when we connect a data UR www.flow.from.one.process to another. Generally become we attach, or pipe, the output of one process to the input of another, ANDLIR



ORANE LABS

ORANE LABS

ORANE

www.oranelabs.com Example ORANE LABS ORANE LABS Lets count the number of files in a directory: NPUR www.oranelabs.com www.o<u>r</u>apelabs.cpm ORANE LABS ORANE LABS ORANE LABS pipe byte stream; stdout stdinls WC (fd 1) unidirectional (fd 0)read end write end OR/ of pipe of pipe SHC HT KANPUR SHC HT KANPUR www.oranelabs.com ORANE LABS ORANE LABS ORANE LABS www.oranelabs.com www.oranelabs.com **Process Pipes** ORANE LABS ORANE LABS Passing data between two programs can be ANPUR www.done using the following functions: www.oranelabs.com ORANE LABS ORANE LABS ORANE LABS #include <stdio.h> FILE \*popen(const char \*command, const char \*open mode); int pclose(FILE \*stream\_to\_close); ORA— The command string is the name of the program ABS to run, together with any parameters. - open\_modemust be either "r"or "w www.oranelabs.com ORANE LABS ORANE LABS ORANE LABS