CS604 MIDTERM CURRENT PAPER 2023 SOLVED BY PRINCE ALVI

1) Subject turnaround time define

Turnaround time: The interval from the time of submission to the time of completion is the turnaround time. Turnaround time is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU and doing I/O. We want to minimize the turnaround time.

2) File description

The file description comprises information concerning the physical structure, identification, and record-names pertaining to a given file. It determines the internal or external attributes of a file connector, of the associated data records, and of the associated data items.

3) Exit system call

A process terminates when it finishes executing its final statement and asks the operating system to delete it by calling the exit system call.

4) Wait system call

The wait () system call

The wait system call suspends the calling process until one of the immediate children terminate, or until a child that is being traced stops because it has hit an event of interest.

5) Why mknod () call fail give two reason.

The mknod () call is normally used for creating special (i.e., device) files but it can be used to create FIFOs too. The 'mode' argument should be permission mode OR-ed with S_IFIFO and 'dev' is set to 0 for creating a FIFO. As is the case with all system calls in UNIX/Linux, mknod () returns –1 on failure and *errno* is set accordingly.

6) CPU scheduler's

Long term

Short term

Middle term

The **Long-term scheduler** (or job scheduler) selects which processes should be brought into the ready queue, from the job pool that is the list of all jobs in the system.

The **Short-term scheduler** (or CPU scheduler) selects which process should be executed next and allocates CPU.

Some operating systems such as time-sharing systems may introduce a **Medium-term scheduler**, which removes processes from memory (and from active contention for the CPU) and thus reduces the degree of multiprogramming.

7) PS – Display status of processes

PS gives a snapshot of the current processes. Without options, PS prints information about processes owned by the user. Some of the commonly used options are -u, -e, and -l.

- -e selects all processes
- -l formats the output in the long format
- -u displays the information in user-oriented format

8) Write system call use to create directory in ~/courses/cs604/
mkdir –p
~/courses/cs604/programs

9) Write given scheduling algorithms in high to low order according to throughput. [1.SJF, 2.RR, 3. Priority queue]
(IDEA SOLUTION)
Throughput: If CPU is busy executing processes then work is being done. One measure of work is the number of processes completed per time, called, **throughput.** We want to maximize the throughput

10) Write the 5 inter communication tools given by Unix/Linux.
UNIX/Linux IPC tools (pipe, named pipe—FIFO, socket, TLI, message queue, shared memory)

11) Five statement option diy hoy thy us ka interrupt, signal and trap batana tha?
An **interrupt** is a signal generated by a hardware device (usually an I/O device) to get CPU's attention.
A **trap** (or an *exception)* is a software-generated interrupt caused either by an error (division by zero or invalid memory access) or by a user request for an operating system service.
A **signal** is an event generated to get attention of a process. An example of a signal is the event that is generated when you run a program and then press <Ctrl-C>.

12) Input command in Linux/ UNIX men batani thi.
command < input-file

13) Write command Moving from background to foreground
Moving a process into foreground
You can use the fg command to resume the execution of a suspended job in the foreground or move a background job into the foreground. Here is the syntax of the command.
fg [%job_id]
where, job_id is the job ID (not process ID) of the suspended or background process. If %job_id is omitted, the current job is assumed.

14) Related to fork ( ) System call
When the fork system call is executed, a new process is created. The original process is called the parent process whereas the process is called the child process. The new process consists of a copy of the address space of the parent. This mechanism allows the parent process to communicate easily with the child process.

15) Scheduling the order in FCFS and Round robin. Scenario given tha
FCFS scheduling algorithm
The process that requests the CPU first (i.e., enters the ready queue first) is allocated the CPU first. The implementation of an FCFS policy is managed with a FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue.

The **round-robin (RR) scheduling algorithm** is designed especially for time-sharing systems. It is similar to FCFS scheduling but preemption is added to switch between processes. A small unit of time, called a **time quantum** (or **time slice**) is defined.

16) Write a command to put process in the background which is already running on foreground

Moving a process into background

you can use the bg command to put the current or a suspended process into the background. Here is the syntax of the command.

bg [%job_id]

If %job_id is omitted the current job is assumed

17) Write 3 standard inputs of pipe () also show their mapping (Aesa e KCH question tha)

- Standard input: 0 (keyboard)
- Standard output: 1 (display screen)
- Standard error: 2 (display screen)

18) Check the errors in the code. Create a thread Wala code tha as it is handouts se. (5)

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
/* Prototype for a function to be passed to our thread */
void* MyThreadFunc(void *arg);
int main()
{
pthread_t aThread;
/* Create a thread and have it run the MyThreadFunction */
pthread_create(&aThread, NULL, MyThreadFunc, NULL);
/* Parent waits for the aThread thread to exit */
pthread_join(aThread, NULL);
printf ("Exiting the main function.\n");
return 0;
}
void* MyThreadFunc(void* arg)
{
printf ("Hello, world! ... The threaded version.\n");
return NULL;
}
```

19) Progress and bounded Waiting

Progress

if no process is executing in its critical section and some processes wish to enter their critical sections, then only those processes that are not executing in their remainder section can

participate in the decision on which will enter its critical section next, and this selection cannot be postponed indefinitely.

Bounded waiting

there exists a bound on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.

20) Threads names

USER AND KERNAL

21) Protection issues

Issues of hardware protection are: I/O protection, memory protection, and CPU protection. We will discuss them one by one, but first we talk about the dual-mode operation of a CPU

22) Difference between Preemptive and non-preemptive scheduling algorithm and which scheduling

Key Differences between Preemptive and Non-Preemptive Scheduling: In preemptive scheduling, the CPU is allocated to the processes for a limited time whereas, in Non-preemptive scheduling, the CPU is allocated to the process till it terminates or switches to the waiting state

23) Similarities between threads and process

Both process and thread are related to each other and quite similar as these are the independent sequence of execution. The basic difference between a process and a thread is that a process takes place in different memory spaces, whereas a thread executes in the same memory space.

24) Differentiate between absolute and relative path

An absolute path is defined as specifying the location of a file or directory from the root directory (/). In other words, we can say that an absolute path is a complete path from start of actual file system from / directory. Relative path is defined as the path related to the present working directly (pwd).

25) Can user process use the kernel code, justify.

User mode processes cannot jump in the kernel unless they use the syscall instruction. It thus allows the kernel to set up some services for user mode that are called system calls

26) Purpose of following commands

Fg
Bg
Ps
Kill
Jobs

1) ps/Function

Report process status

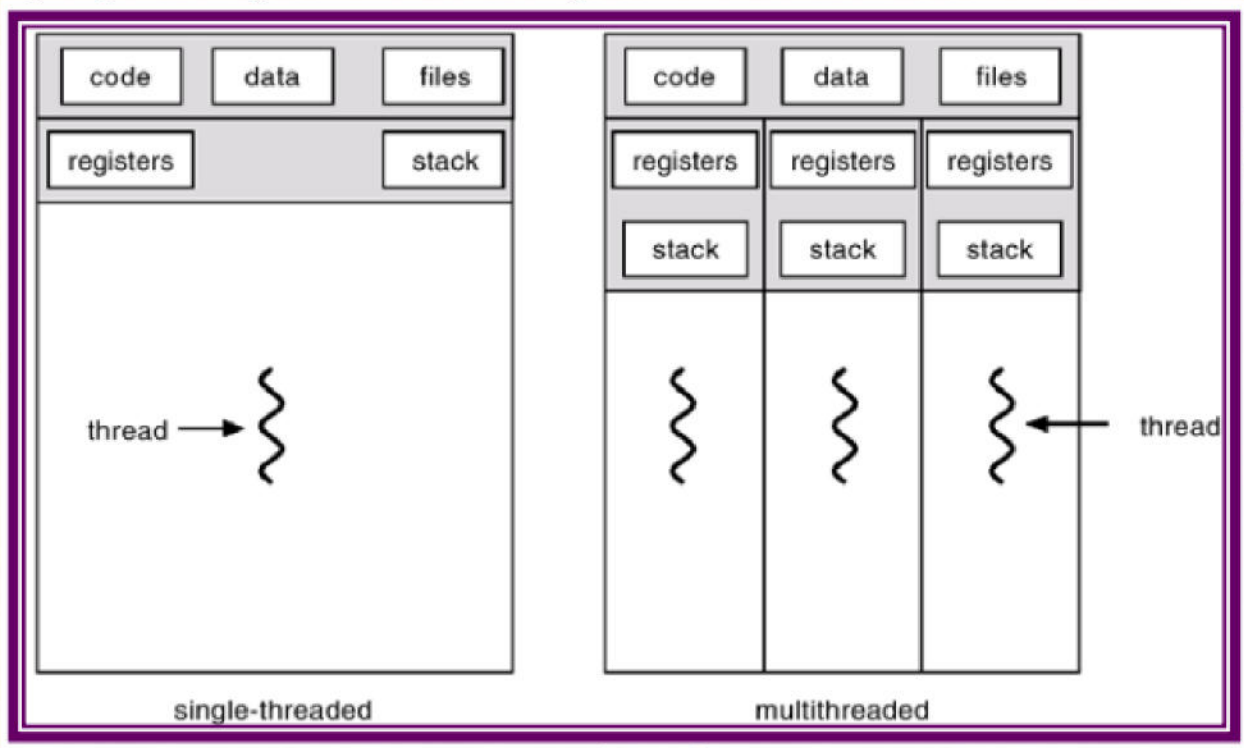
2) Description. If job control is enabled, the fg command moves a background job in the current environment into the foreground. Use the JobID parameter to indicate a specific job to be run in the foreground.
3) bg command in linux is used to place foreground jobs in background.
4) kill/Function
Send the specified signal to the specified process or process group
5) The jobs command will list all jobs on the system; active, stopped, or otherwise.

27) Kernel approached ky advantage...
The benefits of the micro kernel approach include the ease of extending the OS. All new services are added to user space and consequently do not require modification of the kernel. When the kernel does have to be modified, the changes tend to be fewer because the micro kernel is a smaller kernel. The resulting OS is easier to port from one hard ware design to another.

28) Input output aur error ki redirection batani the.
Input Redirection: Here is the syntax for input redirection:
command < input-file
Output Redirection: Here is the syntax for output redirection:
command > output-file
Error Redirection: Here is the syntax for error redirection:
command 2> error-file

29) Advantage of cooperative process
Advantages of Cooperating Process in Operating System
With help of data and information sharing, the processes can be executed with much faster speed and efficiency as processes can access the same files concurrently.

30) Diagram of single and multi-threaded process



31) X and Y want to communicate to each other how many way they can communicate

Cooperating processes require some type of inter-process communication, which is most commonly one of two types: Shared Memory systems or Message Passing systems.

32) High part of os

1) Process Management.
2) I/O Device Management.
3) File Management.

33) Multi level ready queue

Ready Queue: The processes that are residing in main memory and are ready and waiting to execute are kept on a list called the ready queue. This queue is generally stored as a linked list. A ready-queue header contains pointers to the first and final PCBs in the list. Each PCB is extended to include a pointer field that points to the next PCB in the ready queue

34) File descriptor of pip

The pipe () system call creates a pipe and returns two file descriptors, one for reading and second for writing

35) Indirect communication properties

Indirect Communication

With indirect communication, messages can be sent to and received from mailboxes. Here, two processes can communicate only if they share a mailbox. The send and receive primitives are defined as:

- Send (A, message) – send a message to mailbox A
- Receive (A, message) – receive a message from mailbox A.