Shivaji University, Kolhapur Question Bank for Mar 2022 (Summer) Examination

Subject Code: 81547, Subject Name: Operating System – II

| Sr. | 0 4 | A | D | C | D |
|-----|--|-----------------------------|-----------------------------|--------------------------------|-----------------------------|
| No. | Question | A | В | С | D |
| 1 | Pool of internal data buffers are called as | Memory | Free list | Buffer cache | Pool |
| 2 | A is an executable file and a is an instance of the program in execution. | Process, program | Page, segment | Program, process | Application, process |
| 3 | The kernel must write buffer contents to disk before reassigning the buffer this condition is called as | write | delayed write | read | append |
| 4 | A buffer consists of two parts: a that contains data from the disk and a that identifies the buffer. | memory array, buffer header | data array, buffer index | cache array, buffer pointer | buffer header, memory array |
| 5 | A is a file whose data is a sequence of entries, each consisting of an inode number and the name of a file. | Device file | File | directory | folder |
| 6 | The occupies the beginning of a file system, typically the first sector, and may contain the bootstrap code that is read into the machine to boot or initialize, the operating system. | super block | boot block | data blocks | inode list |
| 7 | The is responsible for process synchronization, interprocess communication, memory management, and process scheduling. | system call interface | process control subsystem | file subsystem | hardware control |
| 8 | The algorithm parses the path name one component at a time, converting each component into an inode based on its name and | open | namei | write | read |

| | the directory being searched, and | | | | |
|-----|---|-----------------|---------------|---------------|-------------|
| | eventually returns the inode of the | | | | |
| | input path name. | | | | |
| | 'ialloc' assigns to a newly | | | | None of the |
| 9 | created file. | disk inode | disk block | byte offset | above |
| | Processes can use system | _ | | | |
| 1.0 | call to position the I/O and allow | | | | |
| 10 | random access to the file. | read | creat | mknod | lseek |
| | random access to the me. | | | | |
| | The translates a file | | | | |
| | system address, consisting of a | terminal | disk driver | device driver | |
| 11 | logical device number and block | driver | | | stream |
| | number, to a particular sector on | driver | | | |
| | the disk. | | | | |
| | System call allows a | | | | |
| | process to query the status of file, | | stat | lseek | none of the |
| 12 | returning information such as file | pipe | | | above |
| | type, file owner, file access times, | | | | above |
| | access permissions. | | | | |
| | The system call | | | | |
| | connects the file system in a | | | | |
| 13 | specified section of disk to the | mount | link | unmount | attach |
| | existing file system hierarchy. | • | | | |
| | A process may expand or contract | | | | |
| 14 | its virtual address space with the | sbrk | brk | attachreg | allocreg |
| | system call. | | | | |
| | A process can synchronize its | | | | |
| 15 | execution with the termination of a | fork | wait | exit | close |
| 13 | child process by executing the | | | | |
| | system call. | | | | |
| | When a process accesses a page | | | | |
| 16 | that is not part of its working set, it | validity | protection | invalid | file |
| | incurs a page fault. | | | | |
| | | | stack pointer | | |
| 17 | The register context of a process | processor | and general- | program | all of the |
| | contains | status register | purpose | counter | above |
| | | | register | | |

| Every memory location of a page is addressed by: | | | (Virtual page | | (Virtual page | (page |
|--|-----|---------------------------------------|---------------|---------------|--------------------|---------------|
| addressed by: In UNIX, processes that have finished execution but have not yet had their status collected are known as is the mechanism by which virtual addresses are mapped to physical addresses. 21 Loadreg has which of the following parameter as input: A is contiguous area of virtual address space of a process that can be treated as the distinct object to be shared or protected. 22 In the collection of memory locations that the process can access is called access is called id. 23 After the execution of fork system call, in parent processes of the occurrences of id. 24 A process can synchronize its execution with termination of child process by executing system call. 25 Logical format of consists of four parts, primary headers, section headers, sections are mumber pair in page) pair page) processes pace as processes pace are mapped to physical addresses are mapped process table and process space are again table process space are again table process space are again table process space are are access in a process table and process page are access in a page and process are page are access and address space are access and address space are access and access is called address space are access and access in a page are access and access are access and access are acc | 18 | Every memory location of a page is | number, | Virtual Page | number, byte | number, byte |
| In UNIX, processes that have finished execution but have not yet had their status collected are known as is the mechanism by which virtual addresses are mapped to physical addresses. 21 Loadreg has which of the following parameter as input: 22 Processes that can be treated as the distinct object to be shared or protected. 23 In the collection of memory locations that the process can access is called process table access is called file the process of the occurrences of Signals inform processes of the occurrences of Synchronous 24 A process can synchronize its execution with termination of child process by executing system call. 25 Logical format of consists of four parts, primary headers, section headers, sections are gained and process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in one of the above in the process file in the process file in one of the above in the process file in the process file in one of the above in the process file in the process file in the process file in one of the above in the process file in the proces | | addressed by: | logical page | number | offset in | offset in |
| finished execution but have not yet had their status collected are known as is the mechanism by which virtual addresses are mapped to physical addresses. 21 Loadreg has which of the following parameter as input: 22 A is contiguous area of virtual address space of a process that can be treated as the distinct object to be shared or protected. 23 The collection of memory locations that the process can access is called parent id. 24 After the execution of fork system call, in parent processes of the occurrences of id. 25 Signals inform processes of the occurrences of in in is yet me call. 26 A process can synchronize its execution with termination of child process by executing system call. 27 Loadreg has which of the following processes are mapped to physical addresses are mapped to physical addresses are mapped to physical addresses space of a process table entry | | | number) pair | | page) pair | page) pair |
| had their status collected are known as is the mechanism by which virtual addresses are mapped to physical addresses. Loadreg has which of the following parameter as input: A is contiguous area of virtual address space of a process that can be treated as the distinct object to be shared or protected. The collection of memory locations that the process can access is called process table access is called process the jid. After the execution of fork system call, in parent processes of the occurrences of fid. Signals inform processes of the occurrences of Signals inform processes of the occurrences of Signals inform process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections as file inode file inode file inode file inode above Processes P | | In UNIX, processes that have | | | | |
| had their status collected are known as is the mechanism by which virtual addresses are mapped to physical addresses. 21 Loadreg has which of the following parameter as input: 22 Pointer to region table entry 23 A is contiguous area of virtual address space of a process that can be treated as the distinct object to be shared or protected. 24 The collection of memory locations that the process can access is called process table process table access is called process table call, in parent process, the pid is id. 25 Signals inform processes of the occurrences of for currences of above A process can synchronize its execution with termination of child process by executing system call. 26 Logical format of consists of four parts, primary headers, section headers, sections even as file inode file inode file inode file above 27 Pointer to per process region table entry Pointer to per process region table entry Process table Region Inode file inode file inode file inode file inode file inode above Processes Processes region table entry | 10 | finished execution but have not yet | Sleeping | Stopped | Zombie | Orphan |
| is the mechanism by which virtual addresses are mapped to physical addresses. Pointer to pregion table parameter as input: A is contiguous area of virtual address space of a process that can be treated as the distinct object to be shared or protected. The collection of memory locations that the process can access is called after the execution of fork system call, in parent process, the pid is id. Signals inform processes of the occurrences of in system call. Logical format of consists of four parts, primary headers, section headers, sections Pointer to process region table region table region table entry Process Region type process region table entry Process table Region Inode Process space virtual address space virtual address space wirtual space address space child process space Virtual space address space of the region table entry Virtual space address space of the process space child process space wirtual address space wirtual address space virtual address space wirtual address space wirtual address space of the region table entry Virtual space address space of the region table entry Virtual space address space of the region table entry Virtual space address space of the region table entry Virtual space address space of the region table entry Virtual space address space of the region table entry Virtual space address space of the region table entry Virtual space address space of the region table entry Virtual space address space of the region table entry Virtual space address space of the region table entry Virtual space address spa | 19 | had their status collected are known | processes | Processes | Processes | Processes |
| which virtual addresses are mapped to physical addresses. Pointer to pregion table parameter as input: A is contiguous area of virtual address space of a process that can be treated as the distinct object to be shared or protected. The collection of memory locations that the process can access is called access is called from a call, in parent process, the pid is id. After the execution of fork system call, in parent process of id. Synchronous A process can synchronize its execution with termination of child process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections are mapped or prointer to region table entry Pointer to region table process region table entry Prointer to per process region table entry Process table Region Inode Process space virtual address space virtual address space virtual space address space virtual space address space child process process Child process process 0 None of the above None of the synchronous above Process file inode Process file none of the above | | as | | | | |
| to physical addresses. Dointer to process parameter as input: Pointer to region table entry Pointer to a region table entry Pointer to per process table entry Pointer to per process Pointer to a region table entry Pointer to per process Pointer to a region table entry Pointer to per process Pointer to a region table entry Pointer to a region table entry | | is the mechanism by | | | | |
| Loadreg has which of the following parameter as input: Dointer to region table entry | 20 | which virtual addresses are mapped | Segmentation | Region | Paging | Memory |
| Loadreg has which of the following parameter as input: Consider to region table parameter as input: Pointer to region table entry Pointer to region table entry Pointer to a region table entry Pointer to a region table entry Pointer to a locked region | | to physical addresses. | | | | |
| Loadreg has which of the following parameter as input: region table entry Region type region table entry Process Pointer to a locked region | | | D : 4 4 | | Pointer to per | |
| parameter as input: Control Control Control Control | 21 | Loadreg has which of the following | | D : . | process | Pointer to a |
| A is contiguous area of virtual address space of a process that can be treated as the distinct object to be shared or protected. The collection of memory locations that the process can access is called process table | 21 | parameter as input: | | Region type | region table | locked region |
| of virtual address space of a process that can be treated as the distinct object to be shared or protected. The collection of memory locations that the process can access is called | | | entry | | entry e | |
| that can be treated as the distinct object to be shared or protected. The collection of memory locations that the process can access is called process table process space access is called process table process space access is called process table address space access is called parent call, in parent process, the pid is id. Signals inform processes of the occurrences of Synchronous above A process can synchronize its execution with termination of child process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections access table process table process table process space virtual address space virtual space access table process of virtual space address space virtual space | | A is contiguous area | | | | |
| The collection of memory locations that the process can access is called After the execution of fork system call, in parent processes of the occurrences of A process can synchronize its execution with termination of child process by executing executions with termination of consists of four parts, primary headers, section headers, section headers, section headers, section headers, sections The collection of memory process table process table process space address space virtual space address space child process process of the process of the process of the address space child process of the address space wirtual space address space child process of the address space child process of the above space and process of the above child process of the address space child process of the above space child process of the above child process of the above child process of the above child process of the address space child process of the above c | | of virtual address space of a process | | | | |
| The collection of memory locations that the process can access is called process table process space access is called process table process space access is called parent call, in parent process, the pid is id. Process Process O Signals inform processes of the occurrences of Synchronous A process can synchronize its execution with termination of child process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections The collection of memory process table process space address space wirtual space address space child process of the parent process of the above child process of the above process of the above child process o | 22 | that can be treated as the distinct | Process table | Region | Inode | file table |
| Docations that the process can access is called process table access is process of the access is called process table access is process of the access in process of the access is process of the access in process of the access is process of the access in process of the access is process of the access in process of the access is process of the access in process of the access in process of the access is process of the access in process in pr | | object to be shared or protected. | | | | |
| Docations that the process can access is called process table access is process of the access is called process table access is process of the access in process of the access is process of the access in process of the access is process of the access in process of the access is process of the access in process of the access is process of the access in process of the access in process of the access is process of the access in process in pr | | The collection of moments | | | | |
| After the execution of fork system call, in parent process, the pid is id. Signals inform processes of the occurrences of Synchronous A process can synchronize its execution with termination of child process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections After the execution of fork system parent parent parent parent parent child process parent address space address space wintial space address space address space address space who headers space address s | | - | | | virtual | |
| After the execution of fork system call, in parent process, the pid is id. Signals inform processes of the occurrences of Synchronous A process can synchronize its execution with termination of child process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections After the execution of fork system parent process child process child process of Process 0 None of the above | 23 | | process table | process space | address space | virtual space |
| 24 call, in parent process, the pid is id. parent process child process Process 0 None of the above 25 Signals inform processes of the occurrences of Synchronous Asynchronous synchronous none of the above 26 A process can synchronize its execution with termination of child process by executing exec sleep wait exit 26 Logical format of consists of four parts, primary headers, section headers, sections Executable file inode Process file above | | access is called | | | | |
| 24 can, in parent process, the pid is id. 25 Signals inform processes of the occurrences of Synchronous A process can synchronize its execution with termination of child process by executing system call. 26 Logical format of consists of four parts, primary headers, section headers, sections Executable file inode Child process 0 Asynchronous Asynchronous Asynchronous Synchronous Asynchronous exec sleep wait exit exit Process file none of the above | | After the execution of fork system | | | | |
| Signals inform processes of the occurrences of Synchronous | 24 | call, in parent process, the pid is | - | child process | Process 0 | |
| 25 occurrences of Synchronous Synchronous synchronous synchronous above A process can synchronize its execution with termination of child process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections Executable file inode Process file above | | id. | process | | | above |
| 25 occurrences of Synchronous Synchronous synchronous synchronous above A process can synchronize its execution with termination of child process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections Executable file inode Process file above | | Signals inform processes of the | | | Uni- | |
| A process can synchronize its execution with termination of child process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections Executable file inode Process file above | 25 | | Synchronous | Asynchronous | | |
| execution with termination of child process by executing system call. Logical format of consists of four parts, primary headers, section headers, sections Executable file inode Process file above Process file | | | | | 5, 110111 0110 013 | above |
| 26 process by executing exec sleep wait exit system call. Logical format of consists of four parts, primary headers, section headers, sections Executable file inode Process file above | | A process can synchronize its | | | | |
| system call. Logical format of consists of four parts, primary headers, section headers, sections Executable file file inode Process file above | 2.5 | execution with termination of child | | | •. | |
| Logical format of consists of four parts, primary headers, section headers, sections Executable file inode Process file none of the above | 26 | process by executing | exec | sleep | wait | exit |
| consists of four parts, primary headers, section headers, sections Executable file inode Process file above | | system call. | | | | |
| consists of four parts, primary headers, section headers, sections Executable file inode Process file above | | Logical format of | | | | |
| headers, section headers, sections file file inode Process file above | | _ | E (11 | | | 64 |
| ine above | 27 | | | file inode | Process file | |
| | | | file | | | above |
| <u> </u> | | | | | | |

| 28 | when process executes system call, kernel sets Effective User Id field in the process table and U area to the owner Id of the file. | fork | exec | setgrp | setuid |
|----|--|--------------------|---------------------------|--------------------------------------|----------------------|
| 29 | The scheduler of UNIX belongs to general class of operating system schedulers known as | Round robin | Multilevel Round robin | Round robin with multilevel feedback | Round robin feedback |
| 30 | Process can control the scheduling priority by system call. | decay | nice | priority | random |
| 31 | system call retrieves the cumulative times that the calling process spent executing in user mode and kernel mode. | time | times | stime | timing |
| 32 | Kernel gives measure of how much time system executing in kernel and user mode and how much time it spends in executing individual routines in the kernel. | Monitoring | Accounting | Profiling | Statistics |
| 33 | The clock handler adjusts the priorities of all processes in user mode at second intervals (on System V) and causes the kernel to go through the scheduling algorithm to prevent a process from monopolizing use of the CPU. | 1 | 2 | 5 | 4 |
| 34 | The device is a block device in a configurable section of a disk. | secondary | page | swap | block |
| 35 | have the same function as other drivers to control the transmission of data to and from terminals. | terminal driver | disk driver | device driver | stream |

UNIT – I

- 1. Explain the architecture of UNIX System.
- 2. Draw and Explain Block diagram of UNIX kernel?
- 3. Explain with example Building Block Primitives
- 4. What is a buffer? Explain the structure of Buffer Header.
- 5. Explain an algorithm for Buffer Allocation
- 6. Explain the condition when Kernel wants a particular buffer and that buffer is currently busy.
- 7. Draw and explain Data Structures for File Subsystem
- 8. Explain the bread algorithm?
- 9. Explain the advantages & disadvantages of buffer cache?

UNIT - II

- 1. Explain the algorithm for conversion of pathname to Inode.
- 2. Explain the structure of Regular file.
- 3. What is super block? List and explain various fields of super block?
- 4. What is Inode? Summarize the fields from disk inode?
- 5. If the super block Free Inode list is empty and remembered Inode is 470. Explain the steps to fill the superblock free Inode list.
- 6. Let us assume Disk block contains 1024 bytes and there are 10 direct blocks, 1 single indirect block, 1, double indirect block, 1 triple indirect block. Find the maximum size of the file of a file's table of content. Write your own assumptions if any.
- 7. With the following assumption, find the block number and byte offset of the inode in the block for following inode numbers: 5153, 3015.

Assumptions:

i. Block size: 1024 bytes

ii. Size of disk inode: 64 bytes

iii. Start block of inode list: 500

- 8. Give the fields of in-core copy of Inode.
- 9. Explain iget() algorithm.
- 10. Explain the directories with layout example?

UNIT – III

- 1. Explain the dup() system call with example.
- 2. Explain difference between Named pipe and unnamed pipe.
- 3. Explain the read() system call.
- 4. Write short note on: Change directory and Change root.
- 5. Explain algorithm creat() for creating a new file.
- 6. Draw the file system tree before and after executing following mount() system call.

```
Mount("/dev/dsk1","/usr",0);
```

- 7. Explain the algorithm for mounting a file system?
- 8. Explain the read and write operations in the pipe?
- 9. Explain the algorithm for open system call to open a file?
- 10. Draw and explain the data structures for file system when following system calls are executed:

```
{
    fd1=open("/etc/passwd", O_RDONLY);
    fd2=open("/etc/passwd", O_WRONLY);
    fd3=open("local", O_RDONLY);
    fd4=dup(fd1);
    fd5=dup(fd4);
    close(fd1);
    close(fd3);
}
```

11. Draw and explain the file system data structures for each statement when processes (A/B) executes following system calls:

```
Process A:

fd1=open("/etc/passwd",O_RDONLY);

fd2=open("local",O_RDWR);

fd3=open("/etc/passwd",O_WRONLY);

Process B:

fd1=open("/etc/passwd",O_RDONLY);

fd2=open("private",O_RDONLY);
```

UNIT - IV

- 1. Draw and explain the complete process state transition diagram.
- 2. Write and explain algorithm for allocating a region.
- 3. Explain with diagram the context of a process in detail.

- 4. Explain with example mapping of process virtual address into physical memory address.
- 5. List and explain the fields of process table.
- 6. What is region? Describe algorithm for allocate region?
- 7. What is U area? List fields from the U area?
- 8. What is a region? Discuss mapping between per-process region table and page table.
- 9. Explain with example mapping of process virtual address into physical memory address.
- 10. Write and explain algorithm for allocating a region.
- 11. What is context switch? Explain the steps for Context switch.

UNIT - V

- 1. Explain the algorithm for exit() system call.
- 2. Explain different functions of clock interrupt handler
- 3. Explain system calls for time?
- 4. What is the use of fork system call? Explain the sequence of operations kernel executes for fork?
- 5. What is the use of signal? Explain the types of signals?
- 6. Explain System Boot and the Init process.
- 7. Draw and explain use level and kernel level priority.
- 8. Explain how kernel prevent a process from monopolizing the use of CPU in Unix System V.
- 9. Explain simple process scheduling algorithm with example.
- 10. Explain profiling in detail.
- 11. Explain process scheduling with example.

UNIT-VI

- 1. What is demand paging? Explain data structure used for demand paging?
- 2. Explain the working of page stealer process.
- 3. What is page fault? Explain handling of validity page fault.
- 4. Explain in detail allocation of space on swap device.
- 5. Explain the functions of line discipline and clists?
- 6. Explain the swapping of a process between swap space and main memory?
- 7. Explain the data structures for demand paging?
- 8. Write a short note on: Streams