National University of Computer & Emerging Sciences, Karachi
School of Computing -Spring 2022
Mid-1 Examination
07th March 2022, 08:30 am – 09:30 am

Course Code: CS2006	**Course Name: Operating System**
Instructor Names: Dr. Ghufran Ahmed, Ms. Anaum Hamid, Ms. Tania Erum, Ms. Mubashra Fayyaz, and Ms. Safia Baloch	
Student Roll No:	**Section No:**

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are **4 questions** and **2 pages.**
- All the answers must be solved according to the sequence given in the question paper.

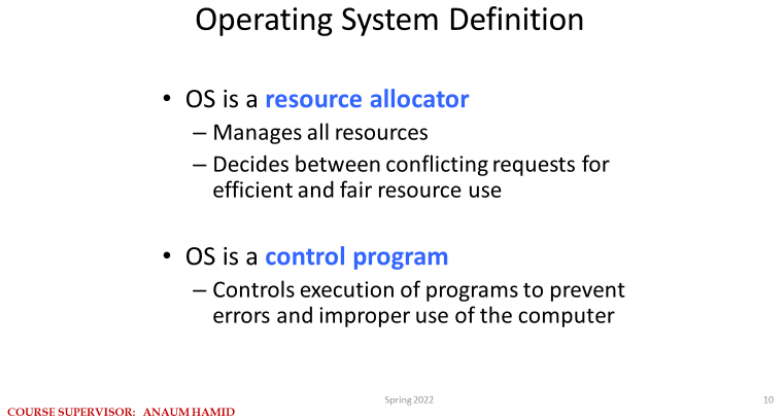
Time: 60 minutes. **Max Marks: 50**

Q1: Introduction to Operating Systems – [Estimated time: 10 mins] [Marks: 5+5+5 = 15]

---

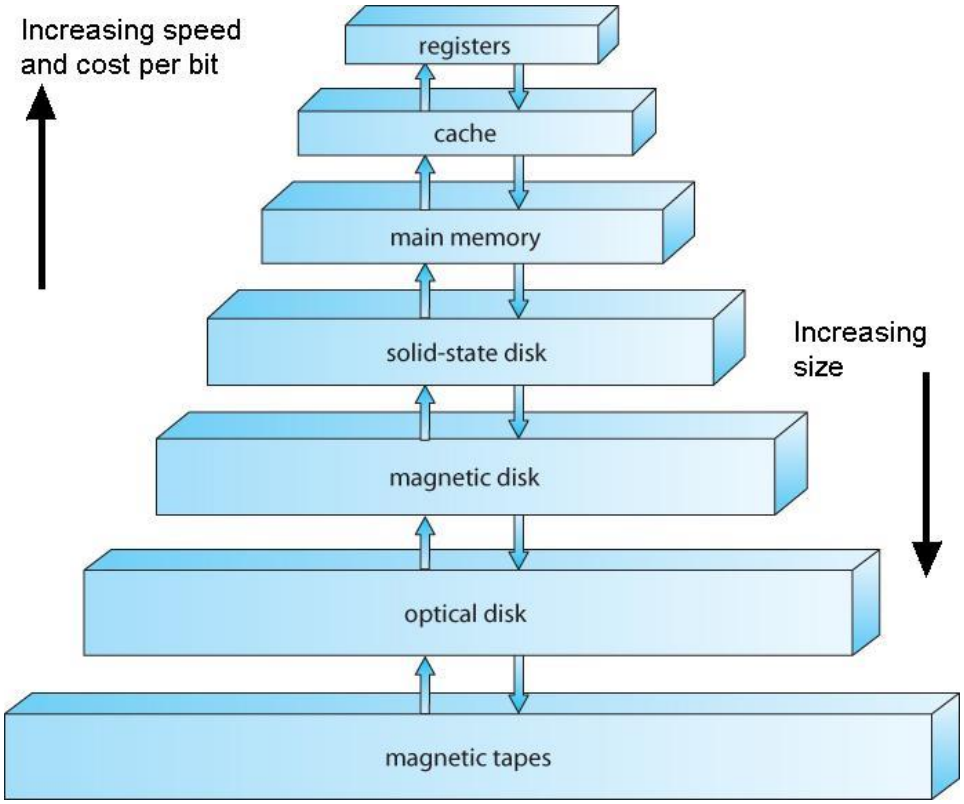
a. Describe the two general roles of an operating system, and elaborate why these roles are important?

The first general role of an operating system is to provide an ABSTRACTION layer for software to run on a machine without needing to know hardware-specific implementation details. It is important in order to reduce the burden on application software developers, extend the basic hardware with added functionality and provide a common base for all applications. The second general role of an operating system is to provide RESOURCE MANAGEMENT to the machine's users, by ensuring progress, fairness and efficient usage of computing resources.



b. Describe different types of storage devices and rank them w.r.t storage and speed. You may support your answer with a properly illustrated diagram.

	1 [2]



c. Define the essential properties of the following types of operating systems:

1. Timesharing
2. Real-time
3. Network
4. Distributed
5. Clustered

Time sharing. **This system uses CPU scheduling and multiprogramming to provide economical interactive use of a system. The CPU switches rapidly from one user to another. Instead of having a job defined by spooled card images, each program reads its next control card from the terminal, and output is normally printed immediately to the screen. d. Real time. Often used in a dedicated application, this system reads information from sensors and must respond within a fixed amount of time to ensure correct performance.**

Real-time. Real-time operating system (RTOS) is an operating system intended to serve real time applications that process data as it comes in, mostly without buffer delay. The full form of RTOS is the Real time operating system.

Network. Provides operating system features across a network such as file sharing. f. SMP. Used in systems where there are multiple CPUs each running the same copy of the operating system.Communication takes place across the system bus.

Distributed.This system distributes computation among several physical processors. The processors do not share memory or a clock. Instead, each processor has its own local memory. They communicate with each other through various communication lines, such as a high-speed bus or local area network.

Clustered. A clustered system combines multiple computers into a single system to perform computational task distributed across the cluster.

Q2: Operating Systems Structures– [Estimated time: 13 mins] [Marks: 5+3+4+3 = 15]

---

	2 [2]

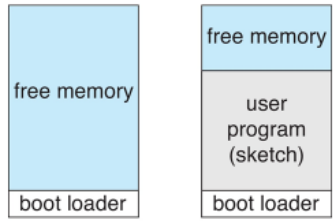
a. How can we distinguish between these terminologies Multiprogramming, Multitasking, and Multiprocessing? Discuss an example of a single-task and multitask Operating System as well.

Multiprogramming means that several programs in different stages of execution are coordinated to run on a single processing unit. **Multiprocessing** is the coordination of the simultaneous execution of several programs running on multiple processing units.**Multitasking** is a logical extension in which the CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing.

Example: Arduino

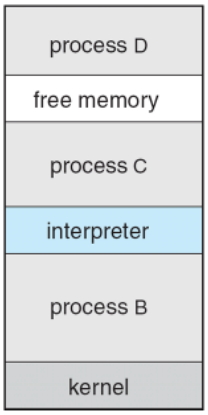
The Arduino is a simple hardware platform consisting of a microcontroller along with input sensors that respond to a variety of events, such as changes to light, temperature, and barometric pressure, etc.

- Single-tasking
- No operating system
- Programs (sketch) loaded via USB into flash memory
- Single memory space
- Boot loader loads program
- Program exit -> shell reloaded



Example: FreeBSD (Berkeley Software Distribution)

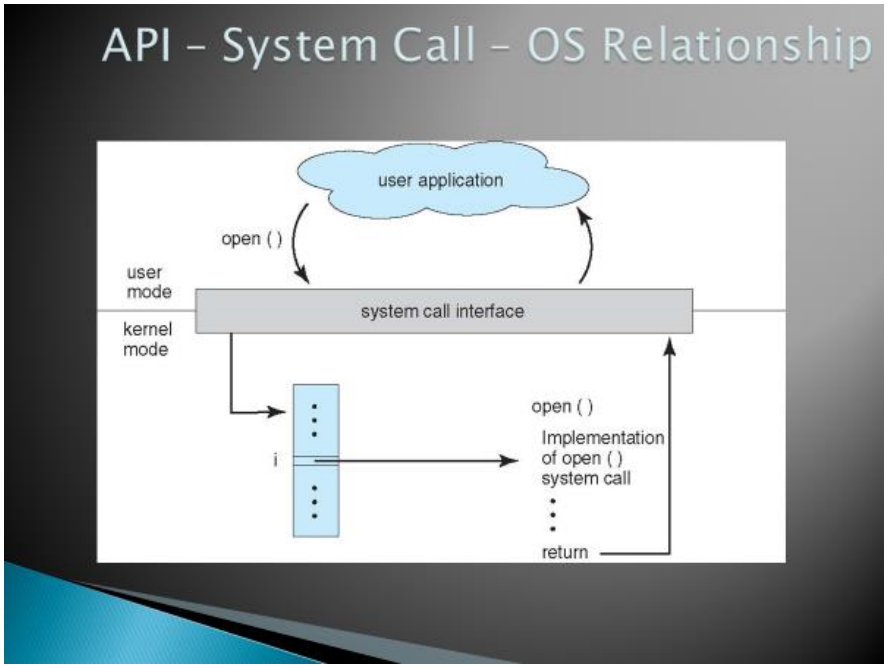
- Unix variant
- Multitasking
- User login -> invoke user's choice of shell
- Shell executes fork() system call to create process
  - Executes exec() to load program into process
  - Shell waits for process to terminate or continues with user commands
- Process exits with:
  - code = 0 - no error
  - code > 0 - error code



b. Consider a system call (e.g., write ()), describing general steps involved in providing the result, from the point of calling the function in the C library to the point where that function returns. You may support your answer with a properly illustrated diagram.

A system call is completed as follows: - As the function is called, an interrupt of the type "software exception" is placed on the processor, causing a Context Switch to take place between the calling function and the kernel. - The exception handler will clear out and save user registers to the kernel stack so that control may be passed on to the C function corresponding to the syscall. - The syscall is executed. - The value(s) returned

by the syscall is placed into the correctly corresponding registers of the CPU (the same ones that a user function normally places its return values in). - The handler takes this value, restores user registers and returns said value to the user programme that called it.



c. What are the three general methods for passing parameters to the operating system? What are the criteria for choosing one of them?

Three general methods used to pass parameters to the OS

1. Simplest: pass the parameters in registers
2. Parameters stored in a block, or table, in memory, and address of block passed as a parameter in a register
3. Parameters placed, or pushed, onto the stack by the program and popped off the stack by the operating system

Criteria:

If there are few parameters to pass: Pass the parameters in registers but this may prove insufficient when there are more parameters than registers. In this case Block and stack methods are used as they do not limit the number or length of parameters passed.

d. Discuss the advantages of operating systems that have loadable kernel modules with an example?

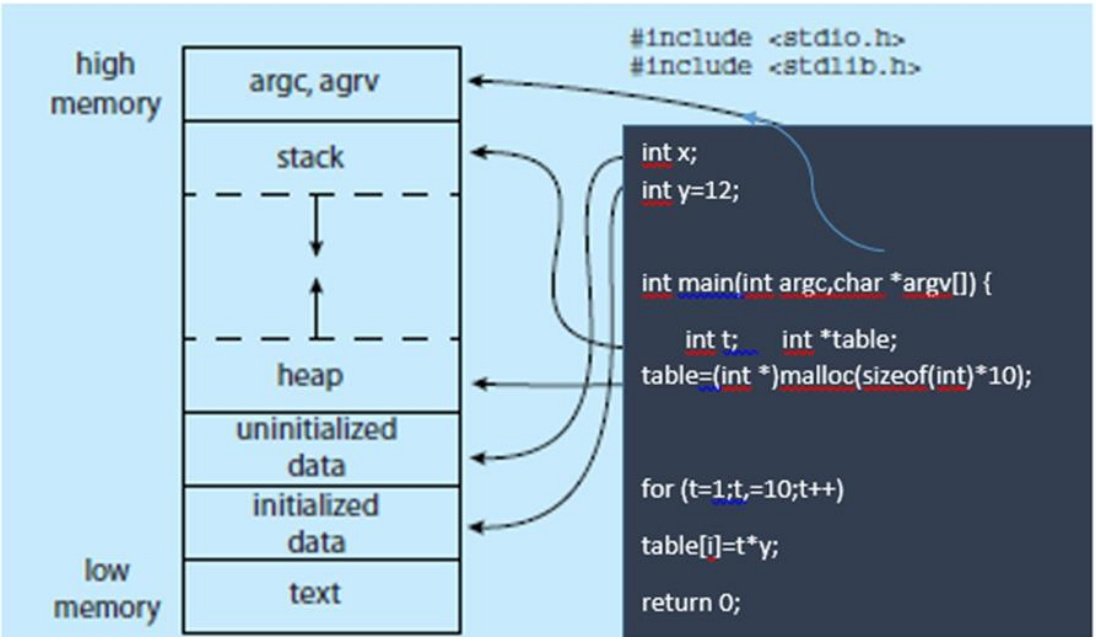
1. **Device drivers don't have to be hard-coded into the kernel**. For example, if a new chip-set comes out that powers many webcams, that kernel module can simply be loaded instead of recompiling the kernel with the new module.
2. It is easier to diagnose system problems.
3. Using modules can save memory, because they are loaded only when the system is actually using them.
4. Modules are much faster to maintain and debug

Q3: Processes – [Estimated time: 10 mins] **[Marks: 5+5 = 10]**

---

a. Draw the memory layout diagram for the following c program:

```c
#include <stdio.h>
#include <stdlib.h>
int x;
int y=12;
int main(int argc, char *argv[]){
    int *table;
    int t;
    table=(int *)malloc(sizeof(int)*10);
        for (t=1;t,=10;t++)
        table[i]=t*y;
    return 0;
}
```



b. Consider the following program code. Answer the following questions:

	5 [2]

```c
#include <sys/types.h>
#include <sys/wait.h>
#include <stdio.h>
#include <unistd.h>
#define SIZE 5
int nums[SIZE] = {0,1,2,3,4};
int main() {
int i;
pid_t pid;
pid = fork();
if (pid == 0) {
        for (i = 0; i < SIZE; i++) {
        nums[i] *= -i;
        printf("%d \n",nums[i]); /* LINE X */
        }
}
else if (pid > 0) {
        wait(NULL);
        execlp("/bin/date","date",NULL);
        printf("%d \n",getpid()); /* LINE Y */
        }
return 0;
}
```

I. Identify ONLY the system call(s) with line number(s).

Line no. 10 fork()

line no. 18 wait()

Line no. 19 execlp()

II. What will be the output of the code?

```
anaum@anaum:~/Documents$ ./test
0
-1
-4
-9
-16
13:56:44 و PKT 2022 مارچ 07 ت
```

III. What will be the output if we remove the wait (NULL) command from line 18?

```
anaum@anaum:~/Documents$ ./test
0
-1
-4
-9
-16
13:56:44 و PKT 2022 مارچ 07 ت
```

IV. Which lines of code will be executed by the child process and the parent process?

Child = 12 to 15

Parent = 18 to 21

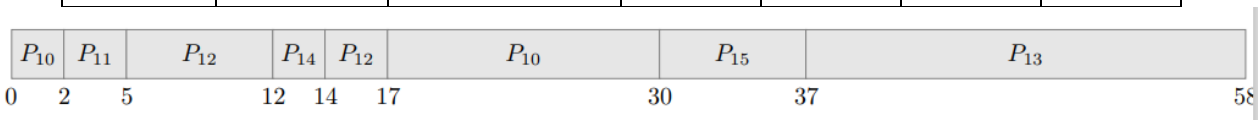
V. When will “LINE Y” be printed?

Line Y never printed because of the execlp() replace the current running process to date command.

Q4: CPU Scheduling – [Estimated time: 12 mins] **[Marks: 5+5 = 10]**

Draw Gantt charts that illustrate the execution of the processes listed in the below Table. Calculate the **average turnaround time** and the **average waiting time** by the **Preemptive** version of the **SJF algorithm**.

Process	Arrival Time	Burst Time	CT	TAT	WT	Priority
P10	0	15	30	30	15	1
P11	2	3	5	3	0	8
P12	5	10	17	12	2	13
P13	11	21	58	47	26	5
P14	12	2	14	2	0	2
P15	25	7	37	12	5	4
				106	48	



Avg WT = 48/6=8
Avg TAT = 106/6=17.66

**************Best of Luck**************