

COS Questions – Lecture 1

Operating System Concepts (Tenth Edition)

Operating System Structures

2.0.1 What is a kernel and what are its objectives?

2.1 What is the purpose of system calls? Give some examples of system calls.

2.2 What is the purpose of the command interpreter? Why is it usually separate from the kernel?

2.3 What system calls have to be executed by a command interpreter or shell in order to start a new process on a UNIX system?

2.4 What is the purpose of system programs?

2.5 What is the main advantage of the layered approach to system design? What are the disadvantages of the layered approach?

2.6 List five services provided by an operating system, and explain how each creates convenience for users. In which cases would it be impossible for user-level programs to provide these services? Explain your answer.

2.10 Describe three general methods for passing parameters to the operating system.

2.12 What are the advantages and disadvantages of using the same system call interface for manipulating both files and devices?

2.13 Would it be possible for the user to develop a new command interpreter using the system-call interface provided by the operating system?

2.14 Describe why Android uses ahead-of-time (AOT) rather than just-in-time (JIT) compilation.

2.15 What are the two models of inter-process communication? What are the strengths and weaknesses of the two approaches?

2.16 Contrast and compare an application programming interface (API) and an application binary interface (ABI).

2.17 Why is the separation of mechanism and policy desirable?

2.19 What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

2.21 How are iOS and Android similar? How are they different?

Process Concept

3.0.1 What is the difference between a process and a program?

3.0.2 What process states are there?

3.0.3 What is the difference between a thread and process?

3.0.4 What is a Process Control Block (PCB) and what is its purpose? Can users access the information stored in one?

3.0.5 What is the difference between heap and stack storage?

3.1 Using the program shown below, explain what the output will be at LINE A.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int value = 5;

int main()
{
    pid_t pid;

    pid = fork();

    if (pid == 0) { /* child process */
        value += 15;
        return 0;
    }
    else if (pid > 0) { /* parent process */
        wait(NULL);
        printf("PARENT: value = %d", value); /* LINE A */
        return 0;
    }
}
```

3.2 Including the initial parent process, how many processes are created by the following program?

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    /* fork a child process */
    fork();

    /* fork another child process */
    fork();

    /* and fork another */
    fork();

    return 0;
}
```

3.3 Original versions of Apple's mobile iOS operating system provided no means of concurrent processing. Discuss three major complications that concurrent processing adds to an operating system.

3.4 Some computer systems provide multiple register sets. Describe what happens when a context switch occurs if the new context is already loaded into one of the register sets. What happens if the new context is in memory rather than in a register set and all the register sets are in use?

3.5 When a process creates a new process using the `fork()` operation, which of the following states is shared between the parent process and the child process?

3.8 Describe the actions taken by a kernel to context-switch between processes.