# Trabajo Práctico 2 - Respuestas hasta el Punto 2

## 1. Responda en forma sintética sobre los siguientes conceptos:

* a. ¿Qué es un Sistema Operativo?

Un Sistema Operativo (SO) es un software que actúa como intermediario entre el hardware y el usuario, gestionando los recursos de la computadora y ofreciendo servicios para la ejecución de programas.

* b. Enumere qué componentes/aspectos del Hardware son necesarios para cumplir los objetivos de un SO.

CPU con modos de ejecución, memoria principal, dispositivos de E/S, reloj del sistema, controladores de interrupciones.

* c. Enumere componentes de un Sistema Operativo

Gestor de procesos, gestor de memoria, sistema de archivos, sistema de E/S, seguridad, gestor de recursos.

* d. ¿Qué es una llamada al sistema (system call)? ¿Cómo es posible implementarlas?

Es el mecanismo mediante el cual los programas solicitan servicios al SO. Se implementan a través de interrupciones o instrucciones especiales que cambian el modo de ejecución a kernel.

* e. Defina y diferencie Programa y Proceso.

Un programa es un conjunto de instrucciones almacenadas en memoria. Un proceso es la ejecución de un programa, con su propio estado, recursos y contexto.

* f. ¿Cuál es la información mínima que el Kernel debe tener sobre un proceso? ¿En qué estructura de datos asociada almacena dicha información?

Debe conocer el estado, contador de programa, registros, memoria asignada, archivos abiertos. Se guarda en el PCB (Process Control Block).

* g. ¿Qué objetivos persiguen los algoritmos de planificación (scheduling).

Optimizar el uso del procesador, maximizar throughput, minimizar tiempos de espera y respuesta, garantizar equidad.

* h. ¿Qué significa que un algoritmo de scheduling sea apropiativo o no apropiativo (Preemptive o Non-Preemptive)?

Preemptive: puede interrumpir un proceso en ejecución para asignar CPU a otro. Non-preemptive: una vez asignada la CPU, el proceso la mantiene hasta finalizar o bloquearse.

* i. ¿Qué tareas realizan los siguientes módulos de planificación?

Short Term Scheduler: selecciona qué proceso en la cola de listos obtiene la CPU.  
Long Term Scheduler: controla la admisión de procesos al sistema.  
Medium Term Scheduler: suspende o reactiva procesos según la carga del sistema.

* j. ¿Qué tareas realiza el Dispatcher? ¿Y el Loader?

Dispatcher: transfiere el control de la CPU al proceso seleccionado (cambio de contexto).  
Loader: carga programas en memoria para su ejecución.

* k. ¿Qué significa que un proceso sea “CPU Bound” y “I/O Bound”?

CPU Bound: requiere principalmente tiempo de CPU. I/O Bound: pasa más tiempo en operaciones de entrada/salida.

* l. ¿Cuáles son los estados posibles por los que puede atravesar un proceso?

Nuevo, Listo, En ejecución, Bloqueado, Terminado. Las transiciones ocurren por admisión, asignación de CPU, bloqueo por I/O, finalización, etc.

* m. ¿Cuáles de los schedulers mencionados anteriormente se encargan de las transiciones entre los estados enumerados?

Short Term Scheduler (Listo → Ejecución), Medium Term Scheduler (Suspensión y reanudación), Long Term Scheduler (Nuevo → Listo).

* n. Defina Tiempo de retorno (TR) y Tiempo de espera (TE) para un proceso.

TR: tiempo total desde la llegada hasta la finalización.  
TE: tiempo total que un proceso permanece en la cola de listos.

* o. Defina Tiempo Promedio de Retorno (TPR) y Tiempo promedio de espera (TPE) para un lote de procesos.

TPR: promedio de los tiempos de retorno de todos los procesos.  
TPE: promedio de los tiempos de espera de todos los procesos.

* p. Defina tiempo de respuesta.

Tiempo transcurrido desde que un proceso llega hasta que se produce la primera respuesta del sistema.

## 2. Algoritmos de Scheduling

### FCFS (First Come First Served)

Funcionamiento: Los procesos se ejecutan en el orden en que llegan, sin interrupciones.

Ejemplo: Si llegan P1(5), P2(3), P3(8), el orden es P1→P2→P3.

Parámetros: No requiere parámetros.

Ventajas: Justo y simple de implementar.

Desventajas: Puede provocar efecto convoy, alto tiempo de espera para procesos cortos.

### SJF (Shortest Job First)

Funcionamiento: El proceso con menor ráfaga de CPU es seleccionado primero.

Ejemplo: Si P1(8), P2(4), P3(2), el orden es P3→P2→P1.

Parámetros: Requiere conocer la duración de la ráfaga (estimada).

Ventajas: Minimiza el tiempo de espera promedio.

Desventajas: Difícil de implementar en la práctica, riesgo de inanición.

### Round Robin

Funcionamiento: Cada proceso recibe la CPU por un tiempo fijo (quantum).

Ejemplo: Con quantum=2 y procesos P1(5), P2(3): ejecución es P1(2)→P2(2)→P1(2)→P2(1)→P1(1).

Parámetros: Quantum (tamaño del tiempo asignado).

Ventajas: Equidad, buena respuesta para sistemas interactivos.

Desventajas: Si el quantum es muy grande se parece a FCFS, si es muy pequeño aumenta el overhead.

### Prioridades

Funcionamiento: Cada proceso tiene una prioridad, se eligen los de mayor prioridad primero.

Ejemplo: Si P1(prio=3), P2(prio=1), P3(prio=2), el orden es P2→P3→P1.

Parámetros: Nivel de prioridad asignado a cada proceso.

Ventajas: Permite dar preferencia a procesos críticos.

Desventajas: Puede causar inanición de procesos con baja prioridad.