

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

3ra práctica (tipo a)
(Primer semestre de 2017)

Horario 0781: prof. V. Khlebnikov

Duración: 1 h. 50 min.

Nota: No se puede usar ningún material de consulta.

La presentación, la ortografía y la gramática influirán en la calificación.

Puntaje total: 20 puntos

Pregunta 1 (2 puntos – 10 min.) Si hay 5 trabajos registrados en este orden y con sus tiempos de ejecución: J1 (7 min.), J2 (11 min.), J3 (4 min.), J4 (20 min.) y J5 (9 min), y si estos trabajos se ejecutan de manera no apropiativa en el mismo orden del registro, entonces los resultados serán obtenidos en 7 min. del trabajo J1, en 18 min. del J2, en 22 min. del J3, en 42 min. del J4 y en 51 min. del J5. Pero, si los trabajos ejecutar de manera apropiativa, según el algoritmo *round-robin* con el *time slice* de 100 milisegundos, ¿cuáles serán los tiempos de terminación de cada trabajo?

Pregunta 2 (4 puntos – 20 min.) Consider the following table with the arrival time and execute time of the process with the quantum time of 100ms:

Process name	Arrival time	Execute time
P0	0	250
P1	50	170
P2	130	75
P3	190	100
P4	210	130
P5	350	50

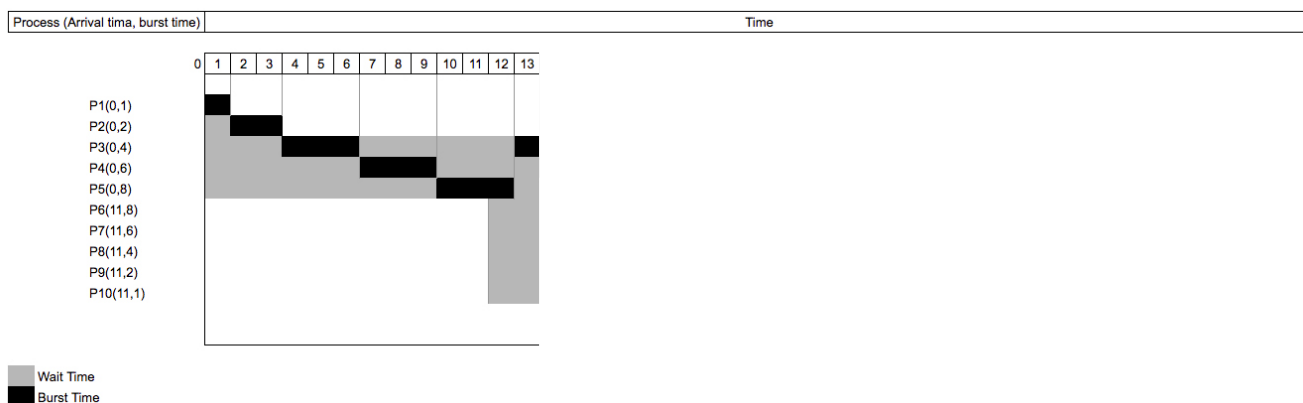
Complete la siguiente tabla hasta que terminen todos los procesos (el subíndice de cada proceso indica el tiempo *faltante* de ejecución, por ejemplo, en el tiempo 50 tenemos P0₂₀₀):

Execute Time

Round Robin Scheduling

0	P0 ₂₅₀	P0 arrives and the gets processed
50	P0 ₂₀₀ P1 ₁₇₀	P1 arrives and waits for quantum to expires
100	P1 ₁₇₀ P0 ₁₀₀	Quantum time 100ms expires, so P0 is forced out of CPU and P1 gets processed
130	P1 ₁₄₀ P0 ₁₀₀ P2 ₇₅	P2 arrives
190	P1 ₁₀ P0 ₁₀₀ P2 ₇₅ P3 ₁₀₀	P3 arrives

Pregunta 3 (4 puntos – 20 min.) Complete la siguiente tabla hasta que terminen todos los procesos: P1(0,1), P2(0,2), P3(0,4), P4(0,6), P5(0,8), P6(11,8), P7(11,6), P8(11,4), P9(11,2), P10(11,1).



Pregunta 4 (1 punto – 5 min.) If the definition is:

$\text{External_Memory_Fragmentation} = 1 - (\text{Largest_Block_Of_Free_Memory} / \text{Total_Free_Memory})$,
fragmentation of 0% means that ... ; fragmentation is 90% (for example) when ... **(complete lo marcado con "...")**.

Pregunta 5 (4 puntos – 20 min.) Considere el siguiente programa en C++ y presente el resultado de su ejecución:

```
#include <bits/stdc++.h>
using namespace std;

void algorithm(int blockSize[], int m, int processSize[], int n)
{
    int allocation[n];          // Stores block id of the block allocated to a process

    memset(allocation, -1, sizeof(allocation));    // Initially no block is assigned to any process

    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (blockSize[j] >= processSize[i]) {
                allocation[i] = j;    // allocate block j to p[i] process
                blockSize[j] -= processSize[i];
                break;
            }
        }
    }
    cout << "\nProcess No.\tProcess Size\tBlock no.\n";
    for (int i = 0; i < n; i++) {
        cout << "    " << i+1 << "\t\t" << processSize[i] << "\t\t";
        if (allocation[i] != -1)
            cout << allocation[i] + 1;
        else
            cout << "Not Allocated";
        cout << endl;
    }
}

// Driver code
int main() {
    int blockSize[] = {100, 500, 200, 300, 600};
    int processSize[] = {212, 417, 112, 426};
    int m = sizeof(blockSize)/sizeof(blockSize[0]);
    int n = sizeof(processSize)/sizeof(processSize[0]);

    algorithm(blockSize, m, processSize, n);

    return 0;
}
```

Pregunta 6 (2 puntos – 10 min.) Usando los datos de la pregunta anterior y la forma de manejo de bloques, presente la tabla con la aplicación de los algoritmos de manejo de memoria que usted conoce. ¿Cuál(es) de ellos proporciona(n) el uso más eficiente de la memoria?

Pregunta 7 (3 puntos – 15 min.) La memoria de 16K (libre inicialmente) se maneja con el algoritmo *buddy system* para los siguientes eventos: el proceso A solicita 3.6K, B solicita 1.5K, C solicita 1.2K, D solicita 1.9K, E solicita 2.7K, C termina, B termina, F solicita 1.5K, G solicita 1.6K, D termina, A termina, G termina, H solicita 6.8K, E termina, I solicita 850 bytes, J solicita 610 bytes, K solicita 1.6K, L solicita 750 bytes, H termina, M solicita 3.9K, N solicita 1.7K, L termina, F termina, K termina, P solicita 670 bytes, I termina, N termina, J termina.
Considere que de los nodos del mismo tamaño siempre se asigna el más a la izquierda. Presente los estados de la memoria manejada.



La práctica ha sido preparada por VK
en Linux Mint 18.1 Serena con LibreOffice Writer.

Profesor del curso: (0781) V. Khlebnikov

Pando, 2 de junio de 2017