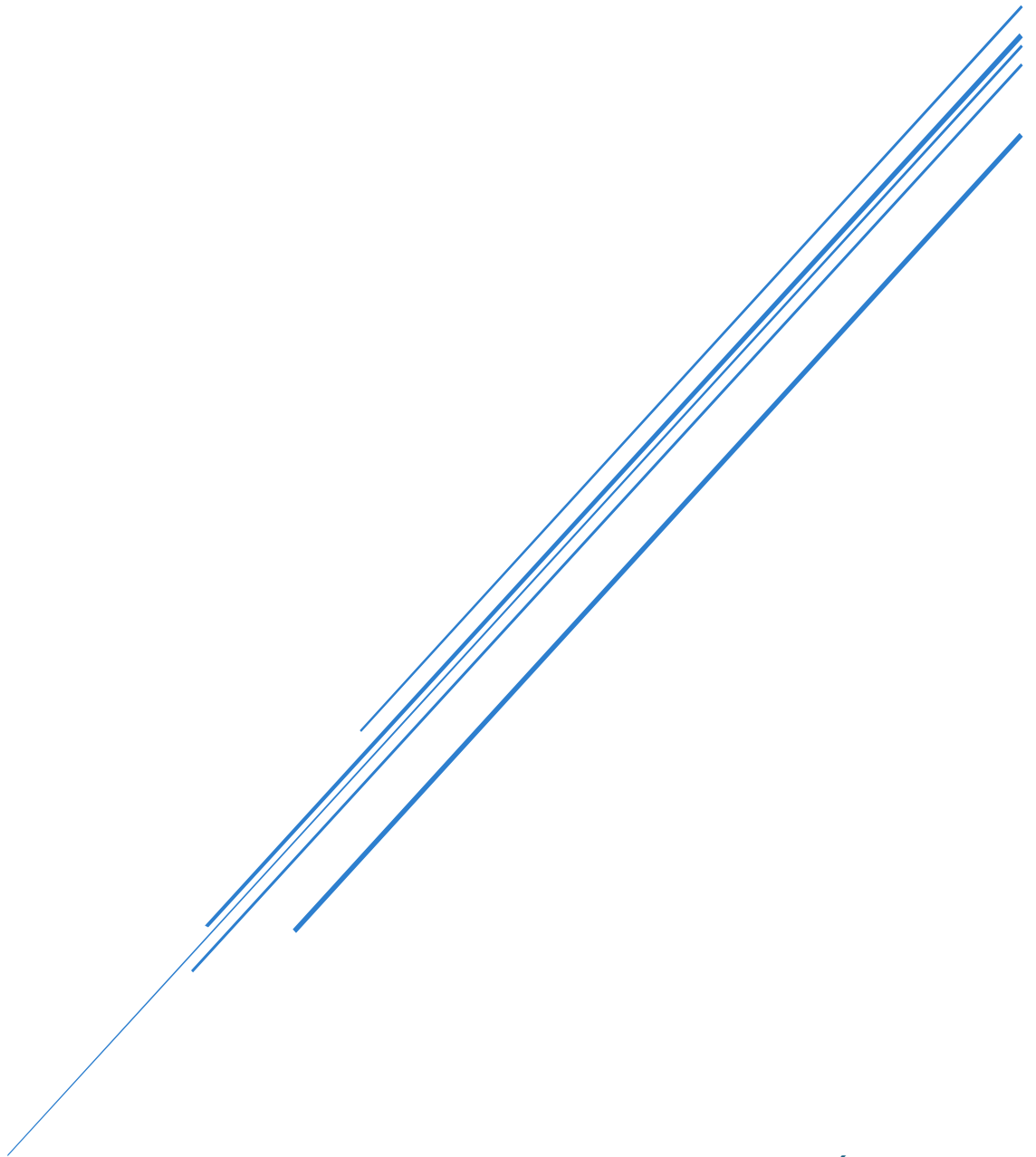


PRÁCTICA 3

SISTEMAS OPERATIVOS



JAIME HERNÁNDEZ
UNIVERSIDAD DE ALICANTE

Se pretende realizar una simulación de gestión de memoria con particiones dinámicas. Se ofrecerá un menú con las dos opciones o algoritmos de asignación de particiones a implementar: primer hueco y mejor hueco.

En este programa Java, implementamos una simulación de asignación de memoria para procesos. En la clase ElementoFichero, encapsulamos la información de un proceso, permitiéndonos crear instancias a partir de valores específicos o de una línea de texto. En nuestra clase principal, gestionprocesos, utilizamos un ArrayList llamado listaFichero para almacenar procesos que esperan ser asignados a la memoria, y una instancia de la clase Memoria para gestionar la asignación propiamente dicha. hueco y siguiente hueco.

El método leerFichero(String nombreFichero) se encarga de leer un archivo de texto con información sobre procesos y llenar la lista listaFichero. Cada línea del archivo se convierte en un objeto ElementoFichero y se agrega a la lista.

```
public static void leerFichero(String nombreFichero) throws FileNotFoundException{
    FileReader fr = null;
    BufferedReader br = null;
    String linea = null;

    fr = new FileReader(nombreFichero);
    br = new BufferedReader(fr);
    try{
        linea = br.readLine();

        while(linea != null && !linea.equals("")){

            listaFichero.add(new ElementoFichero(linea));
            linea = br.readLine();
        }
    }
}
```

```
public static void menu(){
    System.out.println("1. Primer hueco");
    System.out.println("2. Mejor hueco");
    System.out.print("Opcion: ");
}
```

El método simular() ejecuta la simulación de asignación de memoria en diferentes instantes de tiempo. Utilizamos la opción elegida por los usuarios para decidir entre dos estrategias de asignación: "Primer hueco" o "Mejor hueco". Mostramos en la consola la evolución de la memoria en cada instante de tiempo.

```

public static void simular(){
    int instante;
    boolean insertado = false;
    File file = null;

    instante = 1;
    while(!listaFichero.isEmpty() || !memoria.isEmpty()){

        for(int i = 0; i < listaFichero.size(); i++){
            if(listaFichero.get(i).getInstanteLlegada() <= instante){
                if(opcion == 1){

                    insertado = memoria.siguienteHueco(listaFichero.get(i));
                }
                else{

                    insertado = memoria.peorHueco(listaFichero.get(i));
                }

                if(insertado){
                    listaFichero.remove(i);
                    i--;
                }
            }
        }
    }
}

```

En nuestro método principal main(String[] args), utilizamos un objeto Scanner para que los usuarios elijan una opción válida del menú. Si se proporciona un argumento de línea de comandos, llamamos a leerFichero para cargar la información del archivo en listaFichero. Posteriormente, llamamos a simular para ejecutar la simulación.

```

public static void main(String [] args){
    Scanner sc = null;

    sc = new Scanner(System.in);
    do{
        menu();
        opcion = sc.nextInt();
        if(opcion < 1 || opcion > 2)
            System.out.println("Error en la opcion.");
    }while(opcion < 1 || opcion > 2);
    if(args.length == 1){
        try{
            leerFichero(args[0]);
            simular();
        }
        catch(FileNotFoundException e){
            System.out.println("Error al abrir el fichero");
        }
    }
}
}

```

El resultado nos quedará de la siguiente manera:

```
PS C:\Users\ramaj\OneDrive - UNIVERSIDAD ALICANTE\ESTUDIOS\INGENIERIA INFORMÁTICA\2A SO\Prácticas SO\p3-jhd3\src> java GestionProcesos ent.txt
1. Primer hueco
2. Mejor hueco
Opcion: 1
1 [0 P1 300] [300 hueco 1700]
2 [0 P1 300] [300 P2 400] [700 hueco 1300]
3 [0 P1 300] [300 P2 400] [700 hueco 1300]
4 [0 hueco 300] [300 P2 400] [700 hueco 1300]
5 [0 hueco 300] [300 P2 400] [700 P3 200] [900 hueco 1100]
6 [0 hueco 300] [300 P2 400] [700 P3 200] [900 P4 500] [1400 hueco 600]
7 [0 hueco 300] [300 P2 400] [700 P3 200] [900 P4 500] [1400 P5 100] [1500 hueco 500]
8 [0 hueco 300] [300 P2 400] [700 P3 200] [900 P4 500] [1400 P5 100] [1500 hueco 500]
9 [0 hueco 300] [300 P2 400] [700 P3 200] [900 P4 500] [1400 P5 100] [1500 hueco 500]
10 [0 hueco 700] [700 P3 200] [900 P4 500] [1400 P5 100] [1500 hueco 500]
11 [0 hueco 700] [700 P3 200] [900 P4 500] [1400 P5 100] [1500 hueco 500]
12 [0 hueco 700] [700 P3 200] [900 P4 500] [1400 P5 100] [1500 hueco 500]
13 [0 hueco 900] [900 P4 500] [1400 P5 100] [1500 hueco 500]
14 [0 hueco 900] [900 P4 500] [1400 P5 100] [1500 hueco 500]
15 [0 hueco 900] [900 P4 500] [1400 P5 100] [1500 P6 300] [1800 hueco 200]
16 [0 hueco 1400] [1400 P5 100] [1500 P6 300] [1800 hueco 200]
17 [0 hueco 1400] [1400 P5 100] [1500 P6 300] [1800 hueco 200]
18 [0 hueco 1400] [1400 P5 100] [1500 P6 300] [1800 hueco 200]
19 [0 hueco 1500] [1500 P6 300] [1800 hueco 200]
20 [0 hueco 2000]
PS C:\Users\ramaj\OneDrive - UNIVERSIDAD ALICANTE\ESTUDIOS\INGENIERIA INFORMÁTICA\2A SO\Prácticas SO\p3-jhd3\src>
```