

Práctica Teórica 1: Introducción

2021 – Sistemas Operativos II

Licenciatura en Ciencias de la Computación

Entrega: martes 23 de marzo

1. Introducción

Indique con cuales de los conceptos de sistemas operativos (abstracción, aislamiento o administración) se vinculan los siguiente items. Un ítem puede relacionarse con varios conceptos, explique brevemente.

1. sockets
2. memoria virtual
3. malloc
4. señales (al estilo de alarm(), el Control-C o kill)
5. gestion de permisos y usuarios
6. bloqueos
7. archivos

2. Estudiando sistemas operativos

Indique dos buenas razones para estudiar sistemas operativos desde el punto de vista de un desarrollador (o programador) y desde el punto de vista de un administrador de sistemas.

3. En el mundo real

Explique algún caso en el que un algoritmo visto en sistemas operativos I se puede aplicar a problemas del mundo real o de la vida cotidiana.

4. Nachos

1. ¿En qué difiere Nachos del sistema operativo que uno usa en su PC?
2. ¿Qué arquitectura de CPU simula Nachos?
3. Investigue brevemente sobre dicha arquitectura ¿de qué tipo es (CISC/RISC)? ¿qué registros tiene? ¿qué otras características destacables puede encontrarle?

4. El siguiente es el código del problema del jardín ornamental implementado con corrutinas, en donde se han forzado cambios de contexto para hacer bien visible el problema. Modifíquelo de forma que no se produzca el problema. El código completo para compilar el programa está disponible en la **Sección Código**.

```
#include "guindows.h"
#include <stdbool.h>
#include <stdio.h>

#define ITERS_PER_TURNSTILE 20

static task t1, t2, tmain, tsched;
static bool t1_done, t2_done;
static unsigned count;

static void ftsched(void)
{
    static unsigned i;
    for (;;) {
        if (i++ % 2 != 0 && !t1_done) {
            ACTIVATE(t1);
        } else if (!t2_done) {
            ACTIVATE(t2);
        } else {
            ACTIVATE(tmain);
        }
    }
}

static void ft1(void)
{
    for (unsigned i = 0; i < ITERS_PER_TURNSTILE; i++) {
        unsigned c = count;
        YIELD(t1);
        count = c + 1;
    }
    t1_done = true;
    YIELD(t1);
}

static void ft2(void)
{
    for (unsigned i = 0; i < ITERS_PER_TURNSTILE; i++) {
        unsigned c = count;
        YIELD(t2);
        count = c + 1;
    }
    t2_done = true;
    YIELD(t2);
}

int main(void)
{
    stack(t1, ft1);
```

```
    stack(t2, ft2);
    stack(tsched, ftsched);
    TRANSFER(tmain, tsched);
    printf("Total count: %u (should be %u)\n",
           count, 2 * ITERS_PER_TURNSTILE);
    return 0;
}
```