

# Examen Mejoramiento Programación de Sistemas

1er Término 2018

Nombre: \_\_\_\_\_ Paralelo: \_\_\_\_\_

## Pregunta 1 (20 puntos)

El siguiente programa tiene una falla de segmentación al ejecutarse debido a un error en el manejo de memoria. Repare el error y muestra la salida correcta del programa.

```
typedef struct node {
    int val;
    struct node * next;
} node_t;

void push(node_t ** head, int val) {
    node_t * new_node;
    Agregar aquí: new_node = (node_t *) malloc(sizeof(node_t));
    (10 puntos)
    new_node->val = val;
    new_node->next = *head;
    *head = new_node;
}

void print_list(node_t * head) {
    node_t * current = head;
    while (current != NULL) {
        printf("%d\n", current->val);
        current = current->next;
    }
}

void main()
{
    node_t *inicio = malloc(sizeof(node_t));
    inicio->val = 10;
    inicio->next = NULL;
    push(&inicio, 5);
    push(&inicio, 4);
    push(&inicio, 3);
    push(&inicio, 2);
    print_list(inicio);
}
```

**Salida: (10 puntos)**

**2**

3  
4  
5  
10

## Pregunta 2 (30 puntos)

Considere un dispositivo embebido con 1024 bytes de memoria, little endian. Dado el siguiente código, escriba el diagrama del contenido de la memoria, byte por byte, en hexadecimal, incluyendo las direcciones. Asuma que las variables se van guardando de manera consecutiva, empezando desde dirección 0x1A.

```
short a[] = {28,32,44};  
char w = 42;  
int *p = &a[2];  
int c = *p;
```

Asumiendo arquitectura de 32 bits:

1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28
1C	00	20	00	2C	00	2A	1E	00	00	00	2C	00	00	00

Asumiendo arquitectura de 64 bits:

1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C
1C	00	20	00	2C	00	2A	1E	00	00	00	00	00	00	00	2C	00	00	00

Direcciones están correctas: 10 puntos

Contenido de memoria está correcto: 15 puntos

Todo está convertido a hexadecimal correctamente: 5 puntos

## Pregunta 3 (25 puntos)

Asumiendo que el siguiente código ha sido compilado correctamente, al ejecutar el código habrá un error en tiempo de ejecución. Encuentre el error en el código y repárelo, explique porque es un error y muestre la salida del programa una vez que el error ha sido reparado.

```
#define N 5
```

```
//Retorna puntero a 0 si *ptr es un número primo  
void *es_primo(void *ptr)  
{
```

```
int *n = (int *) ptr;
int *r = (int *) calloc(1, sizeof(int));
```

Se debe eliminar la línea de abajo porque el hilo no puede ser independiente debido a que tiene que ser esperado después por el hilo principal con `pthread_join`. (Error correcto: 5 puntos, Explicación correcta: 10 puntos)

```
pthread_detach(pthread_self());
for(int i=2;i < ((int) sqrt(*n) + 1);i++)
    if(*n%i == 0){
        *r = i;
        return (void *) r;
    }

return (void *) r;
}

int main()
{
    int numeros[N] = {100,13,25,49,11};
    int *resultados[N];
    pthread_t tid[N];
    int i;

    for(i=0;i<N;i++)
        pthread_create (tid + i, NULL, es_primo, numeros + i);

    for(i=0;i<N;i++)
        pthread_join(tid[i], (void **) (resultados + i));

    for(i=0;i<N;i++)
        printf("%d ", *(resultados[i]));
}
```

Salida: (10 puntos)

2 0 5 7 0

## Pregunta 4 (25 puntos)

Considere el siguiente código. ¿Cuál es el contenido final del archivo a.txt?

```
int main(void){
    int fd = open("a.txt", O_CREAT | O_RDWR); //fd = 3
    int fd2 = open("a.txt", O_TRUNC | O_RDWR); // fd2 = 4
    write(fd, "yay", 3);
    printf("%d\n",res);
    write(fd2, "nay",3);
    int fd3 = open("a.txt", O_APPEND | O_RDWR);
    write(fd3, "hey", 3);
    char a[10] = {0};
    snprintf(a,10, "%d", fd3); //número a string fd3 = 5
    write(fd3, a, strlen(a)); //strlen --> tamaño de string
    close(fd);
    close(fd2);
    close(fd3);
    return 0;
}
```

Contenido correcto:

**nayhey5**

**Si muestra "nay": 5 puntos**

**Si muestra "hey": 5 puntos**

**Si muestra "5" al final: 10 puntos**

**Si muestra "yay" en el contenido: -10 puntos**

**Si muestra "res" en el contenido: -15 puntos**