

Implementación de Técnicas de Gestión de Memoria en C

Esta actividad consta de 3 puntos donde los conceptos de gestión de memoria en sistemas operativos se aplican mediante implementaciones en C. Cada punto aborda una técnica diferente de gestión de memoria.

Punto 1: Implementación de Paginación

Implementa un programa en C que simule la paginación. El programa debe:

- *Inicializar una tabla de páginas con un tamaño definido (por ejemplo, 256 páginas).
- **Asignar páginas a marcos físicos (por ejemplo, 64 marcos) de forma dinámica.
- ***Manejar accesos a memoria e imprimir fallos de página cuando una página no está asignada.

Instrucciones:

- Define las estructuras necesarias para la tabla de páginas y la memoria física.
- Implementa funciones para inicializar la tabla de páginas, asignar páginas a marcos y acceder a la memoria.
- Simula varios accesos a memoria y muestra los resultados en la consola.

Ejemplo de uso:

```
#include <stdio.h>
#include <stdlib.h>
```

```
#define PAGE_SIZE 4096
#define NUM_PAGES 256
#define FRAME_SIZE 4096
#define NUM_FRAMES 64
```

```
int page_table[NUM_PAGES];
int memory[NUM_FRAMES][FRAME_SIZE];
```

```
void initialize_page_table();
int allocate_page(int page_number);
void access_memory(int page_number, int offset);
```

```

int main() {
    initialize_page_table();
    allocate_page(1);
    access_memory(1, 100);
    access_memory(2, 50);
    return 0;
}

```

Punto 2: Implementación de Segmentación

Implementa un programa en C que simule la segmentación. El programa debe:

1. Inicializar una tabla de segmentos.
2. Asignar segmentos con una base y un límite específicos.
3. Manejar accesos a memoria e imprimir fallos de segmentación cuando el acceso esté fuera de los límites del segmento.

Instrucciones:

- Define la estructura para un segmento.
- Implementa funciones para inicializar la tabla de segmentos, asignar segmentos y acceder a la memoria.
- Simula varios accesos a memoria y muestra los resultados en la consola.

Ejemplo de uso:

```

#include <stdio.h>
#include <stdlib.h>

#define MAX_SEGMENTS 10

typedef struct {
    int base;
    int limit;
} Segment;

Segment segment_table[MAX_SEGMENTS];

void initialize_segment_table();

```

```

int allocate_segment(int base, int limit);
void access_memory(int segment_number, int offset);

int main() {
    initialize_segment_table();
    allocate_segment(1000, 500);
    access_memory(0, 100);
    access_memory(0, 600);
    return 0;
}

```

Punto 3: Implementación de Swapping

Implementa un programa en C que simule el swapping. El programa debe:

1. Definir un área de memoria principal y un área de swap.
2. Implementar funciones para swap-in y swap-out de procesos entre la memoria principal y el área de swap.
3. Simular el proceso de swapping e imprimir el estado de la memoria antes y después de la operación.

Instrucciones:

- Define las estructuras para la memoria principal y el área de swap.
- Implementa funciones para realizar swap-in y swap-out.
- Simula varios procesos de swapping y muestra los resultados en la consola.

Ejemplo de uso:

```

#include <stdio.h>
#include <stdlib.h>

#define MEMORY_SIZE 1024
#define SWAP_SIZE 2048

int memory[MEMORY_SIZE];
int swap[SWAP_SIZE];

void swap_out(int process_id, int start, int size);
void swap_in(int process_id, int start, int size);

int main() {
    swap_out(1, 0, 512);
}

```

```
    swap_in(1, 0, 512);  
    return 0;  
}
```

Solución Esperada

1. Punto 1: Paginación

- El programa debe inicializar la tabla de páginas, asignar páginas y manejar accesos a memoria con fallos de página.

2. Punto 2: Segmentación

- El programa debe inicializar la tabla de segmentos, asignar segmentos y manejar accesos a memoria con fallos de segmentación.

3. Punto 3: Swapping

- El programa debe realizar swap-in y swap-out de procesos, mostrando el estado de la memoria antes y después de la operación.

Entrega

- Cada implementación debe ser entregada como un archivo .c separado.
- Asegurar de incluir comentarios en el código para explicar el funcionamiento de cada parte.
- Probar el código y muestra ejemplos de salida en la consola.