

1 Objetivos

1. Entender el concepto de "PageFault" y comprender los elementos involucrados.
2. Entender el concepto de Swap
3. entender la estructura de un TLB y manejar las excepciones de páginas faltantes

2 pageFault

1. Estudiar el método "Translate" e identificar y anotar en su diseño cómo y cuándo es que se produce el "PageFault exception"

- (a) para ello debe revisar el código en el archivo "machine/translate.cc"

La excepción de "pagefault" se da cuando ocurre una de las 2 siguientes situaciones: se intenta acceder a una página que es más grande que la TLB o se intenta acceder a una página que no se encuentra válida.

- (b) Para realizar las lecturas o escrituras a la memoria del procesador MIPS, el simulador utiliza los métodos "Machine::ReadMem" y "Machine::WriteMem", note que estos métodos utilizan direcciones virtuales o lógicas, anote ¿Por qué?; explicar los casos en que estos métodos retornan falso.

Este método solamente retorna falso cuando existe una excepción.

- (c) Explicar como se traducen las direcciones virtuales a físicas en los métodos anteriores. Para esto primeramente se obtiene el valor de la página y el desplazamiento, a partir del valor de la página se obtiene el "pageframe" desde la TLB o la pagetabl, finalmente se obtiene la dirección física a partir de la siguiente línea de código

```
1 *physAddr = pageFrame * PageSize + offset;
```

- (d) Explicar el funcionamiento del método "Translate", describa los parámetros que recibe este método, su tipo y una explicación propia de para qué se utilizan, además anote el tipo de retorno del método, las posibles excepciones que puede retornar y por qué se generan.

Primeramente podemos ver en la siguiente línea de código los parametros del metodo:

```
1 Machine::Translate(int virtAddr, int* physAddr, int size, bool writing)
```

El primer parametro es la dirección de memoria virtual que es manejada por la maquina MIPS, el segundo es un apuntador a un entero que apuntara a la dirección de memoria fisica del sistema linux, el tercer parametro es la cantidad de bytes que se escribirán (1,2 o 4) y finalmente el ultimo parametro indica si se realizará una lectura o escritura a memoria

2. Describir la manera cómo se emplea el TLB (Translation Look-Aside Buffer) en el procesador MIPS.

Esta se emplea como un TranslationEntry de solamente 4 entradas para simular un cache.

- (a) Describir en su diseño los elementos de la estructura "pageTable" en la clase "AddrSpace".

La estructura pagetable se conforma de un arreglo de "translationEntrys", esta clase posee atributos como dirección física, dirección virtual, uso, estado entre otros.

- (b) Explicar lo que hacen los métodos "RestoreState" y "SaveState" de la clase "AddrSpace" Estos métodos se encargan de reestablecer el estado de la "machine" y guardarlo, el método de restablecer como tal, no se encuentra implementado en la versión inicial.

- (c) Anotar la descripción de la declaración de "tlb" en "machine.h"

La tlb funciona de manera parecida a la pagetable solamente que esta tiene un máximo de 4 entradas.

- (d) Explicar la inicialización de la variable "tlb" en "machine.cc"

Esta inicialización lo que pretende es liberar la tlb y dejar limpia para su uso posterior, además de fijar el tamaño a 4 entradas. Dado que esta no puede ser cambiada

- (e) Anotar los elementos de la estructura "tlb", explique porqué son iguales a los elementos de "pageTable" de "AddrSpace"

```
1 TranslationEntry *tlb;    // this pointer should be considered
2    // "read-only" to Nachos kernel code
3
4 TranslationEntry *pageTable;
```

Como se puede observar en el extracto de machine.h ambas estructuras son apuntadores de translationEntry.

- (f) Explicar para qué sirven las variables "pageTable" y "tlb" en la clase "Machine" Estas estructuras son utilizadas para poder generar un cambio en el contexto, dado que Machine es el hardware y estas variables simulan el hardware del procesador con respecto al manejo de páginas.

- (g) Indicar cómo se realiza la búsqueda de la dirección virtual y anote cómo se calcula la página que la contiene, la variable "vpn"

El valor del vpn se calcula mediante la dirección virtual dada dividida entre la cantidad de páginas. Esto separa la página del offset que quedaría como el residuo.

```
1 vpn = (unsigned) virtAddr / PageSize;
```

- i. Anote el procedimiento cuando el simulador utiliza la variable "pageTable" para realizar la búsqueda

```
1 if (vpn >= pageTableSize) {
2     DEBUG('a', "virtual page # %d too large for page table size %d!\n",
3         virtAddr, pageTableSize);
4     return AddressErrorException;
5 } else if (!pageTable[vpn].valid) {
6     DEBUG('a', "virtual page # %d too large for page table size %d!\n",
7         virtAddr, pageTableSize);
8     return PageFaultException;
9 }
10 entry = &pageTable[vpn];
```

ii. Anote el procedimiento cuando se utiliza la variable "tlb"

```
1 for (entry = NULL, i = 0; i < TLBSize; i++)
2     if (tlb[i].valid && (tlb[i].virtualPage == vpn)) {
3         entry = &tlb[i];        // FOUND!
4         break;
5     }
6 }
```

iii. Si la página es válida, en ambos casos se define una variable "entry", anote qué representa y cuál es su contenido lógico.

Esta variable es una instancia de "pageentry" y contiene la dirección física y virtual entre otras propiedades mencionadas anteriormente

3. Indicar cómo se asocia la página virtual al marco de memoria física correspondiente

(a) Explicar que representa la variable "pageFrame" en el método "Translate".

La variable pageframe representa la página en donde se debe buscar la dirección ya que cuando se lee de memoria se realiza por página y no una sola dirección de memoria.

(b) Explicar cómo se obtiene la dirección física y cómo la devuelve este método "Translate". Para esto primeramente se obtiene el valor de la página y el offset, a partir del valor de la página se obtiene el pageframe desde la TLB o la pagetabl, finalmente se obtiene la dirección física a partir de la siguiente línea de código

```
1 *physAddr = pageFrame * PageSize + offset;
```

(c) Explicar que hace esta porción de código del método "Translate", indique en cuál estructura de datos ocurren estos cambios.

```
1 ...
2 entry->use = true;    // set the use, dirty bits
3 if (writing)
4     entry->dirty = true;
5 ...
```

Estos cambios se realizan sobre pageentry, en donde se actualiza el uso de la página y además si se realiza una escritura se modifica el valor de dirty para indicar que esta página ha sido modificada.

(d) Explicar la funcionalidad de la bandera (bit) "dirty".

Si se realiza una escritura se modifica el valor de "dirty" para indicar que esta página ha sido modificada.

4. Identificar los casos en que se produce el "PageFaultException" en el método "Translate"

(a) El simulador de MIPS no avanza los contadores de programa cuando ocurren las excepciones, en el caso de "SysCallException" lo tuvimos que hacer 'manualmente', en el caso de "PageFaultException" no hay que hacerlo, pues es indispensable que la instrucción que causó la excepción sea reejecutada.

(b) Explicar cómo va a resolver cada uno de los casos encontrados en que ocurre una excepción "PageFault"

Lo que se pretende realizar en estos casos es cargar en la memoria del tlb la página solicitada y volver el contador a la posición anterior para que la vuelva a correr.

5. Explicar cómo va a obtener las páginas de la memoria principal para asignarlas a las páginas faltantes.

- Explicar cómo va a resolver el problema si toda la memoria física está llena.
Cuando la memoria física esta llena se reliza un reemplazo de la página que fue utilizada de último con el algoritmo fifo.
- Explicar en que casos la página que se reemplaza debe ir a SWAP y en que caso no.
En los casos en que la memoria ha sido modificada debe ir al swap y en caso contrario no.
- Analizar todos los casos posibles para la página que necesita ser reemplazada y su diseño de solución.

6. Utilización del archivo de intercambio (SWAP)

- (a) Explicar la estructura interna del archivo de intercambio SWAP.
El archivo esta compuesto por varios bitmaps los cuales guardan la información de las páginas virtuales y las páginas físicas
- (b) Indicar el momento en que el archivo debe construirse y destruirse.
El archivo de swap se crea al crearse una nueva instancia de memoria virtual ya que este esta contenido en el constructor de dicha clase y este a su vez es llamado desde la creación del addres space
- (c) Indicar la cantidad de elementos que este archivo debe contener.
Por regla general (dada por el profesor) esta zona debe contener el doble de páginas que la memoria de pagetables osea 64 páginas de 128 bytes.
- (d) Explicar su estrategia para que NachOS conozca cuáles de esos elementos están ocupados y cuáles están libres.
Mediante el uso de bitmaps se puede lograr saber cuales están libres y cuales no.

3 Características de la página faltante

1. Determinar cuál es la página de la dirección lógica que generó la CPU, es decir, la que provocó la excepción (PageFault exception)

- (a) Explicar por qué se generó esta excepción.
Esta excepción se genera porque no todas las páginas del programa son cargadas al inicio, solamente las páginas que se utilizan son las que se cargan.
- (b) Determinar cuál es la dirección que causó la excepción. Para ello debe revisar "machine.h" y la definición de los registros de la CPU, en particular ver el registro 39.

```
1 #define BadVAddrReg 39 // The failing virtual address on an exception
```

Como se muestra en el código anterior, en el registro 39 se guarda la memoria que ocasiona una excepción de manejo de memoria virtual, por lo que al leerlo se obtiene dicha dirección.

- (c) Explicar la manera en que se calcula la porción de la dirección lógica correspondiente a la página y la porción para el desplazamiento.
La página se calcula al dividir entre la constante pagesize y el desplazamiento al obtener el módulo entre dicha constante.

- (d) Indicar el procedimiento a seguir para lograr que no vuelva a ocurrir esa excepción de falta de página.

Para que no vuelva a ocurrir dicha excepción se debe de cargar a memoria cache o tlb la página completa donde se ocasionó la excepción.

2. Propiedades de la página faltante

- (a) Describir el procedimiento a seguir si la página hay que leerla del archivo original (ejecutable). Piense en la posibilidad de que el archivo deba ser abierto nuevamente y que para ello es necesario que su NachOS conozca el nombre del archivo, esta sería una nueva característica de la clase "AddrSpace".

- (b) Analizar el caso en el que la página se encuentra sucia, anotar el procedimiento para recuperarla.

cuando una página se encuentre sucia se debe guardar a memoria principal para que se encuentren sincronizadas y de esta manera pasar al estado "clean", así poder ser leída desde memoria sin ningún problema.

- (c) De acuerdo con las características de la página faltante descritas, construir una tabla de casos e indicar la estrategia para recuperar la página faltante en cada caso.

- i. Si se trata de una página de Pila, S , Datos inicializados, Di , Datos No inicializados , Dni o de Código, T
- ii. Si está sucia o no, explicar cuáles páginas no se pueden ensuciar y por qué.
Una página puede estar sucia cuando se realiza una escritura a la misma y se modifican datos en la página. las páginas con el segmento de código no se pueden ensuciar porque son de solo lectura.
- iii. Si la página va de salida (víctima) o de entrada (solicitada)
- iv. De donde la página es tomada o donde es colocada.
La colocación de la página está dada por el bitmap y si se encuentra lleno se debe liberar la última página utilizada.

4 Actualización de estructuras

1. Explique qué método cambia las banderas (bits) "dirty" y "use" y en cuál estructura de datos ocurre ese cambio.

La bandera de dirty es cambiada cuando se realiza escrituras en la página y la bandera de use se cambia cuando una página es cargada a memoria.

2. Anotar los casos en que se debe cambiar el bit de validez de una página (de falso a verdadero y viceversa), indique cuales estructuras de datos deben reflejar ese cambio.

En el address space se cambia la validez de una página al ser cargada se pasa a verdadero y cuando se desecha la página se debe cambiar a falso. al eliminar los hilos de un programa las páginas utilizadas pasan a un estado de validez falso

3. Indicar cuáles estructuras debe actualizar para evitar que esa excepción ("PageFaultException") siga ocurriendo.

se deben realizar cambios en el tlb de tal manera que se guarde la página que se intentó acceder y causó la excepción

4. En los métodos "RestoreState" y "SaveState" indicar los cambios que considera necesarios para mantener el funcionamiento correcto de la memoria virtual de NachOS.

Se debe realizar la implementación del restore state de tal manera que se cargen las páginas del tlb a pagetable del address space

- (a) En NachOS cuando ocurre un cambio de contexto, el hilo que sale llama al método "SaveState" y el hilo que entra llama a "RestoreState"

- (b) Recuerde que al igual que con "pageTable" de la clase "Machine", solo vamos a tener un "tlb" que utilizan todos los hilos, el hardware modifica el TLB, pero esos cambios deben persistir en las "pageTables" de cada hilo.

Para lograr esto, se debe realizar una copia del tlb de machine al pagetable del hilo de addresspace, de esta manera se garantiza que ambos poseen las mismas páginas

- (c) Explicar qué debe hacer NachOS cuando tiene que sacar un hilo de la máquina para cederlo a otro, explicar qué cambios pueden ocurrir en los datos de la TLB y en qué lugar se debe almacenar esos cambios.

Cuando se realiza un swap entre 2 hilos, las páginas utilizadas por el hilo se deben de guardar en el archivo de swap, de esta manera se pueden intercambiar 2 hilos sin perder las páginas que estaban utilizando. si una página se encuentra sucia, esta debe ser guardada en memoria principal.

- (d) Explicar los cambios necesarios a aplicar en los métodos anteriores para permitir ese cambio de contexto, indique cual sería el estado correcto de la TLB de la máquina cuando un proceso recién comienza a correr.

Para esto se debe modificar la creación de la page table de tal manera que no se utilice ni se cargue a memoria las páginas del programa, solamente se crea el page table y se rellena de falso, además se debe modificar la clase pageentry de tal manera que se pueda tener un índice para la zona de swap.