Bedienungsanleitung des Projektes Compilieren/Bedienen

Konfiguration und Bedienung

Alle häufig genutzten Einstellungen sind zentral auffindbar in *main.cpp*:

1. TLB-Größe: *tlb_size*;
   Steuert die Kapazität des TLB

2. Anzahl physischer Frames: *frame_count*;
   Bestimmt wie viele Seiten gleichzeitig im physischen Speicher liegen dürfen. Bei Vollauslastung wird der gewählte Paging-Algorithmus zur Opferwahl aufgerufen.

3. Virtueller Adressraum: *proc1(int process_id, int virtual_pages_count)*;
   Passt die Größe der Seitentabelle des Prozesses an

Algorithmus auswählen

Um einen Algorithmus auszuwählen, muss das jeweilige Algorithmus-Objekt in *main.cpp* angelegt werden. Die Algorithmen NRU, FIFO, SecondChance, LRU, NFU und NFUAging sind schon angelegt. Um einen Algorithmus zu wählen, müssen die jeweils anderen Objekte auskommentiert werden.

Hinweis: Manche Algorithmen benötigen Zugriff auf die Seitentabelle, deshalb wird ihnen *proc1.getPageTable()* übergeben

Eventfolge einstellen

Für die verschiedenen Algorithmen kann man Eventfolgen erstellen, z.B.:

```
void createQueue_FIFO(des::EventQueue& queue) {

using namespace des;

...

}
```

Die Eventfolge baut man dann mithilfe von

```
queue.push({timestamp, EventType::PAGE_ACCESS, page_id});
```

Die EventQueue wird dann mit PAGE_ACCESS-Ereignissen und Timestamps befüllt:

createQueue_FIFO(queue);

Mithilfe von Auskommentieren kann die jeweilige Eventfolge gewählt werden.

Ausgaben

- **[EVENT] ...**: Beginn/Ende eines Seitenzugriffs samt Timestamp.
- **[MMU] ...**: Treffer/Miss in TLB und Seitentabelle; Laden/Entladen von Seiten; Seitenfehler.
- **[TLB] ...**: Invalidate-Hinweise, wenn ein Opfer aus dem TLB entfernt wird.
- **[ALGO] ...**: Algorithmus-Meldungen.
- **Statistik**: Gibt eine Statistik über die Anzahl an Seitenzugriffen, TLB-Treffer, sowie Seitenfehler an.

Mögliche Erweiterungen

- Algorithmen: Es können weitere Algorithmen hinzugefügt werden. Dafür muss *on_page_access(int), on_page_fault(int)* und *select_victim()* implementiert werden, falls der Algorithmus Seitentabellen-Infos benötigt, muss im Konstruktor eine *PageTable&*-Referenz hinzugefügt werden
- Eventtypen: Um weitere Eventtypen hinzuzufügen muss *EventType* in *discrete_event_simulation.h* erweitert werden. Zusätzlich muss die *MMU::handle_event()*-Logik angepasst werden.
- Mehrprozessorbetrieb: Erweiterung durch mehrere Prozesse mit jeweils eigener Seitentabelle, eigenem virtuellen Adressraum und eigenem PID.
- Statistikwerte: Durch hinzufügen weiterer Zähler könnten weitere Statistiken hinzugefügt werden. Außerdem wäre es möglich, dass man zur Verbesserung der Aussagekraft der Statistik mehrere Page-Access-Queues mit jeweils unterschiedlichen Zugriffssequenzen durchläuft. Für jede Queue berechnen wir dann das Verhältnis aus der Anzahl der Page-Faults und der Gesamtzahl der Zugriffe (Page-Faults / Page-Accesses) und ermitteln im Anschluss den Durchschnitt all dieser Verhältnisse. Auf diese Weise wird der Einfluss der konkreten Zugriffsreihenfolge vollständig eliminiert.