Rozkazy asemblerowe, interpreter oraz programy testowe

Lab. systemy operacyjne

Kamil Korbik

Krótki opis mechanizmu działania interpretera

1. Działanie interpretera rozpoczyna moduł interfejsu, który wywołuje metodę `void KK_Interpret()` służącą do zinterpretowania oraz wykonania jednego rozkazu asemblerowego zapisanego w pliku txt.

2. metoda `KK_Interpret()` uruchamia inne metody służące do pobrania zawartości zmiennej globalnej PCB i umieszczenia jej w zmiennych wewnętrznych.

3. Odczytuje licznik rozkazów aktywnego procesu, po czym pobiera dwa bajty rozkazu od pamięci wirtualnej, zwiększając również licznik rozkazów o dwa.

4. Następnie rozpoznaje rozkaz i sprawdzając w mapie, ile potrzebuje argumentów, pobiera po jednym bajcie, aż nie natrafi na spację.

5. Po otrzymaniu parametrów, wykonuje przypisane do rozkazu instrukcje.

6. Następnie zwiększa wartość licznika rozkazów o ilość bajtów wykorzystanych do rozkazu.

7. W kolejnym kroku zapisuje wszystkie wartości zmiennych pobranych z PCB z powrotem do zmiennej.

8. Oddaje kontrolę do modułu interfejsu.

Przykładowe działanie

1. Z PCB procesu zostaje odczytany stan licznika rozkazów oraz stany rejestrów, które wynoszą 0.
2. Metoda `KK_Interpret` z wykorzystaniem metody modułu pamięci wirtualnej, po podaniu jej adresu rozkazu, pobiera dwa bajty – „AD” i licznik rozkazów zwiększany jest o 2.
3. Interpreter sprawdza w mapie, że wysłany rozkaz korzysta z dwóch parametrów i oczekuje ich do znaku spacji, ponieważ nie wie ile bajtów zajmują.
4. Otrzymuje parametr AX oraz 15
5. Następnie wykonuje zaprogramowane dla AD działania, mające na celu dodanie do rejestru AX wartości 15.
6. Po wykonaniu rozkazu zapisuje dane (Licznik = 8, AX = 15) do PCB i oddaje sterowanie do interfejsu, który może ponownie wywołać funkcję `KK_Inretpret()` dla tego samego procesu, bądź dla innego z innym PCB.

Struktury danych

- Wartości rejestrów typu int
- Wartość licznika rozkazów typu short
- Mapa ilości parametrów danego rozkazu
- Parametry rozkazu typu string
- Vector stringów zapisujący parametry

Główne funkcje i procedury

- `Void KK_Interpret()` – najważniejsza funkcja wywołująca interpreter dla procesu w stanie `RUNNING`.
- Funkcje poszczególnych rozkazów np.: `void KK_mult(Vector<string> a).`
- `Void KK_DispReg()` – wyświetlanie zawartości rejestrów.
- `String KK_ByteFromMem()` – pobieranie bajtów z pamięci.

Wywoływanie metod

- Metoda `KK_Interpret` oraz `KK_DispReg` są metodami publicznymi i mogą być wykorzystywane przez inne moduły systemu operacyjnego.
- Funkcja `KK_Interpret` wywoływana jest przez interfejs w celu rozpoczęcia lub kontynuowania interpretacji rozkazów.
- Funkcja `KK_DispReg` może być wywołana przez interfejs w celu podejrzenia, jakie wartości znajdują się w rejestrach.

Rozkazy asemblera

- Arytmetyczne:
  - AD `rejestr :: liczba || rejestr :: adres || rejestr :: rejestr` – dodawanie
  - SU `rejestr :: liczba || rejestr :: adres || rejestr :: rejestr` – odejmowanie
  - MU `rejestr :: liczba || rejestr :: adres || rejestr :: rejestr` - mnożenie
  - IN `adres || rejestr` – inkrementacja
  - DE `adres || rejestr` – dekrementacja
- Skoki:
  - JU `adres` – skok bezwarunkowy
  - JZ `adres` – skok jeśli rejestr równy jest zero
  - JN `adres` – skok jeśli rejestr jest różny od zera
  - SP – zakończenie programu

Rozkazy asemblera

- Przenoszenie
  - MO `rejestr :: liczba || rejestr :: adres || rejestr :: rejestr` - przenoszenie do :: z
- Działanie na plikach
  - OF `nazwa ::  adres :: tryb (0,1)` – otwórz plik (0 – tryb odczytu, 1 – zapisu)
  - WF `adres pliku :: adres pierwszego bajtu danych :: ilość bajtów do zapisania` – zapisanie do pliku
  - RF `adres pliku :: adres pierwszego bajtu danych do odczytu :: ilość bajtów do odczytu` – odczyt z pliku
  - CF `adres pliku` - zamyka plik
  - DF `adres pliku` – usuwa plik

Rozkazy asemblera

- Operacje na procesach:
  - CP `nazwa procesu :: priorytet` – tworzenie nowego procesu
  - DP `nazwa procesu` – usuwanie procesu