

Risc-V: Implementazione algoritmo di Mergesort

Il Problema

Nella relazione andrò a presentare una soluzione funzionante fornita per implementare l'algoritmo di Mergesort in architettura RISC-V e scritta in Assembly.

L'algoritmo

L'algoritmo di Mergesort è un algoritmo di ordinamento ricorsivo basato sul Divide Et Impera basato su confronti.

L'algoritmo funziona nel seguente modo:

1. Se la sequenza è composta da 0 o 1 numeri allora è già ordinata altrimenti si divide la sequenza in due sottosequenze, se il numero di elementi della sequenza è dispari allora la prima sottosequenza avrà un elemento in più della seconda;
2. L'algoritmo viene applicato ricorsivamente sulle sottosequenze fino a quando entrambe non sono ordinate;
3. A questo punto le due sottosequenze ordinate vengono fuse (*merge*) estraendo ripetutamente il minimo dalle due sottosequenze ponendolo nella sequenza uscente che alla fine risulterà completamente ordinata.

L'algoritmo ha complessità $\Theta(n \log n)$.

L'algoritmo di Mergesort è uno degli algoritmi di ordinamento più famosi in quanto in diverse situazioni si rivela essere ottimale.

La soluzione proposta

La soluzione proposta si suddivide in due file:

- **main.s:** in questo file si concentra la fase preparatoria per il resto del programma, nella sezione `.rodata` viene dichiarata la dimensione dell'array mentre nella sezione `.data` viene dichiarato l'array vero e proprio. Nella sezione `.text` vengono inizializzati i registri che andranno a contenere gli indirizzi del primo e dell'ultimo elemento dell'array.
- **mergesort.s:** in questo file sono contenute le diverse funzioni che andranno ad ordinare la sequenza numerica

- **compile.sh**: script contenente le istruzioni per compilare il codice sorgente

Le funzioni principali

`mergesort`

- È la funzione principale e viene chiamata per la prima volta in `main.s` con l'indirizzo del primo e l'indirizzo dell'ultimo elemento dell'array come parametri
- Calcola il numero di elementi nella sequenza, se è presente un solo elemento allora la funzione termina (jump alla funzione `mergesort_end`) altrimenti:
- Divide il numero di elementi presenti nell'array in due e calcola l'indirizzo dell'elemento a metà
- Chiama ricorsivamente sé stessa due volte:
 - La prima volta sulla sequenza a partire dal primo elemento fino a quello a metà
 - La seconda volta a partire dall'elemento dopo quello a metà fino all'ultimo

`merge`

- Viene chiamata dopo che `mergesort` ha completato le due chiamate ricorsive
- Gli vengono passati come parametri
 - Indirizzo del primo elemento
 - Indirizzo dell'elemento a metà
 - Indirizzo dell'ultimo elemento
- È la funzione che provvede a fondere due sottosequenze ordinate dando come ritorno una sequenza ordinata
- Scorre insieme le due sottosequenze mettendo nella sequenza di ritorno l'elemento più piccolo presente nelle due

Verifica delle funzionalità

Per verificare il funzionamento del programma procediamo nel seguente modo:

- Apriamo due terminali nella cartella contenente i sorgenti
- Eseguiamo in un terminale lo script di compilazione tramite il comando `./compile.sh` che genererà un eseguibile chiamato `mergesort`
- Fatto ciò nello stesso terminale lanciamo `qemu` tramite il comando `qemu-riscv64 -g 2233 mergesort`
- Fatto ciò andiamo nell'altro terminale e colleghiamoci a `qemu` tramite il comando `riscv64-unknown-elf-gdb -ex "tar rem:2233" mergesort`
- A questo punto partirà l'esecuzione del programma, andiamo avanti inserendo `s` nel terminale di `gdb` per vedere i vari passaggi del codice

- Per verificare la riuscita del programma continuiamo ad andare avanti avendo la premura di fermarci poco prima che il programma termini, ad esempio quando raggiungiamo l'istruzione `ecall` e verifichiamo se la sequenza è ordinata inserendo il comando `x/5xb &testArray` (il cinque va sostituito nel caso con un numero diverso in quanto corrisponde alla dimensione dell'array inserito nella sezione dati).

Test con istanze di input

Di seguito alcune immagini che mostrano il funzionamento con tre istanze diverse di input

1) `testArray: .byte 5, 3, 8, 2, 20`

```
calcolatori@calcolatori-VirtualBox: /media/sf_riscv-mergesort
GNU gdb (GDB) 8.0.50.20170724-git
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org
This is free software: you are free to change and redistrib
There is NO WARRANTY, to the extent permitted by law. Type
and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --t
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources onli
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "wor
Reading symbols from mergesort...done.
Remote debugging using :2234
_start () at main.s:17
17      la a0, testArray          # Load testArray a
(gdb) s
18      lb t0, size                # Load the array l
(gdb) x/5xb &testArray
0x11281: 0x05 0x03 0x08 0x02 0x14
(gdb)
```

```
calcolatori@calcolatori-VirtualBox: /media/sf_riscv-mergesort 7
(gdb)
shift_skip () at mergesort.s:78
78      addi s0, s0, 1            # Increment first hal
(gdb)
79      ld a2, 24(sp)             # Load back last elem
(gdb)
81      bge s0, a2, merge_loop_end
(gdb)
merge_loop_end () at mergesort.s:107
107     ld ra, 0(sp)
(gdb)
108     addi sp, sp, 32
(gdb)
merge_loop_end () at mergesort.s:109
109     ret
(gdb)
mergesort_end () at mergesort.s:40
40      addi sp, sp, 32
(gdb)
mergesort_end () at mergesort.s:41
41      ret
(gdb)
_start () at main.s:23
23      ecall
(gdb) x/5xb &testArray
0x11281: 0x02 0x03 0x05 0x08 0x14
(gdb)
```

2) `testArray: .byte 10, 9, 8, 7, 6, 5, 4`

```
calcolatori@calcolatori-VirtualBox: /media/sf_riscv-mergesort 77x28
calcolatori@calcolatori-VirtualBox: /media/sf_riscv-mergesort$ riscv64-unknown
-elf-gdb -ex "tar rem:2234" mergesort
GNU gdb (GDB) 8.0.50.20170724-git
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=riscv64-unkno
wn-elf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from mergesort...done.
Remote debugging using :2234
_start () at main.s:17
17      la a0, testArray          # Load testArray address in a0
(gdb) s
18      lb t0, size                # Load the array length
(gdb) x/7xb &testArray
0x11281: 0x0a 0x09 0x08 0x07 0x06 0x05 0x04
(gdb)
```

```
calcolatori@calcolatori-VirtualBox: /media/sf_riscv-mergesort 77x28
78      addi s0, s0, 1            # Increment first half index and
point to the next element
(gdb)
79      ld a2, 24(sp)             # Load back last element address
(gdb)
81      bge s0, a2, merge_loop_end
(gdb)
82      bge s1, a2, merge_loop_end
(gdb)
merge_loop_end () at mergesort.s:107
107     ld ra, 0(sp)
(gdb)
108     addi sp, sp, 32
(gdb)
merge_loop_end () at mergesort.s:109
109     ret
(gdb)
mergesort_end () at mergesort.s:40
40      addi sp, sp, 32
(gdb)
mergesort_end () at mergesort.s:41
41      ret
(gdb)
_start () at main.s:23
23      ecall
(gdb) x/7xb &testArray
0x11281: 0x04 0x05 0x06 0x07 0x08 0x09 0x0a
(gdb)
```

```
3) testArray: .byte 127, 40, 32, 11, 15, 90, 1, 8
```

```

calcolatori@calcolatori-VirtualBox: /media/sf_riscv-mergesort 70x28
GNU gdb (GDB) 8.0.50.20170724-git
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copy
and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=riscv6
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from mergesort...done.
Remote debugging using :2234
_start () at main.s:17
17      la a0, testArray          # Load testArray address  in a
(gdb) s
18      lb t0, size              # Load the array lenght
(gdb) x/8xb &testArray
0x11281:  0x7f  0x28  0x20  0x0b  0x0f  0x5a  0x01
(gdb)

```

```

calcolatori@calcolatori-VirtualBox: /media/sf_riscv-mergesort 85x28
78      addi s0, s0, 1           # Increment first half index and point to
the next element
(gdb)
79      ld a2, 24(sp)           # Load back last element address
(gdb)
81      bge s0, a2, merge_loop_end
(gdb)
82      bge s1, a2, merge_loop_end
(gdb)
merge_loop_end () at mergesort.s:107
107     ld ra, 0(sp)
(gdb)
108     addi sp, sp, 32
(gdb)
merge_loop_end () at mergesort.s:109
109     ret
(gdb)
mergesort_end () at mergesort.s:40
40      addi sp, sp, 32
(gdb)
mergesort_end () at mergesort.s:41
41     ret
(gdb)
_start () at main.s:23
23     ecall
(gdb) x/8xb &testArray
0x11281:  0x01  0x08  0x0b  0x0f  0x20  0x28  0x5a  0x7f
(gdb)

```