

Architettura degli elaboratori - lezione 7

Appunti di Davide Vella 2024/2025

Claudio Schifanella

claudio.schifanella@unito.it

Link al moodle :

<https://informatica.i-learn.unito.it/course/view.php?id=3106>

18/03/2025

Contenuti

1. [Convenzione nell'uso e salvataggio dei registri](#)
 1. [Spiegazione del problema](#)
 2. [Convenzione \(IMPORTANTE\)](#)
2. [Fasi di invocazione di procedura](#)
 1. [Fase 1 - Pre chiamata del chiamante](#)
 1. [Preparazione degli argomenti \(parametri\) della procedura](#)
 2. [Fase 2 - Invocazione della procedura](#)
 3. [Fase 3 - Prologo lato chiamato](#)
 4. [Fase 4 - Corpo della procedura](#)
 5. [Fase 5 - Epilogo lato chiamato](#)
 6. [Fase 6 - Ritorno al chiamante](#)
 7. [Fase 7 - Post chiamate lato chiamante](#)
3. [Fasi di una invocazione di una procedura foglia](#)
 1. [Fase 1 - Pre chiamata del chiamante](#)
 2. [Fase 2 - invocazione della procedura](#)
 3. [Fase 3 - prologo lato chiamato](#)
 4. [Fase 4 - Corpo della procedura](#)
 5. [Fase 5 - Epilogo lato chiamato](#)
 6. [Fase 6 - ritorno al chiamante](#)
 7. [Fase 7 - post chiamata lato chiamante](#)
4. [Record di attivazione - struttura](#)

Convenzione nell'uso e salvataggio dei registri

Spiegazione del problema

1. Chi è responsabile di salvare i registri quando si effettuano chiamate di funzioni? Dipende :
 - La funzione chiamata conosce i registri che essa userà, quindi prima dell'uso li salva, li usa e poi carica i vecchi valori dei registri.
 - La funzione chiamante conosce quali registri sono importanti per essa, quindi li salva prima di chiamare una funzione.
2. Bisogna evitare le inefficienze, ovvero salvare il minor numero di volte qualcosa nei registri (sw, lw, ...):
 - La funzione chiamante potrebbe salvare tutti i registri che sono importanti per sé, anche se la procedura chiamata non li modificherà.
 - La funzione chiamata potrebbe salvare tutti i registri che si appresta a modificare, anche quelli che non verranno poi utilizzati dalla procedura chiamante una volta che la procedura chiamata le avrà restituito il contenuto.Entrambe queste soluzioni vanno quasi bene, ma non sono ben ottimizzati, allora si può fare un altro modo, la convenzione.

Convenzione (IMPORTANTE)

1. I registri x10-x17 (a0-a7), x5-x7 e x28-x31 (t0-t6):
 - Possono essere modificati dal chiamato senza nessun meccanismo di ripristino.
 - Il chiamante se necessario dovrà salvare i valori dei registri prima dell'invocazione della procedura.
2. I registri x1 (ra), x2(sp), x3(gp), x4(tp), x8(fp/s0), x9 e x18-x27 (s1-s11):
 - Se modificati dal chiamato devono essere salvati e poi ripristinati prima del ritorno al chiamante.
 - Il chiamante non è tenuto al loro salvataggio e ripristino.

Fasi di invocazione di procedura

Fase 1 - Pre chiamata del chiamante

Se uno (o più) dei registri che sono allocati alle procedure (x10-x17 (a0-a7), x5-x7 e x28-x31 (t0-t6)) contiene qualcosa che ci serve, dobbiamo salvarlo prima di chiamare la procedura.

Preparazione degli argomenti (parametri) della procedura

I primi 8 argomenti vengono posti in x10-x17 (a0-a7). Eventuali altri argomenti vanno salvati nello stack (EXTRA_ARGS), così che si trovino subito sopra il frame della funzione che chiamata.

Fase 2 - Invocazione della procedura

Usiamo l'istruzione "jal NOME_PROCEDURA".

Ricorda, "jal" vuol dire "jump and link". Salta alla procedura e poi linka, imposta il PC.

Fase 3 - Prologo lato chiamato

Eventuale allocazione del call-frame sullo stack (aggiornare sp) :

- Salvataggio di x1 (ra) nel caso in cui la procedura non sia foglia (ovvero una procedura che non chiama altre procedure)
- Salvataggio di x8 (fp), solo se utilizzato all'interno della procedura
- Salvataggio di x9 e x18-x27 (s1-s11) se utilizzati all'interno della procedura (il chiamante si aspetta di trovarli intatti)
- Salvataggio degli argomenti x10-x17 (a0-a7) solo se la funzione li riuserà successivamente a ulteriori chiamate a funzione (procedure innestate)
- Eventuale inizializzazione di fp : punta al nuovo call-frame

Fase 4 - Corpo della procedura

La procedura fa quello che deve fare.

Fase 5 - Epilogo lato chiamato

1. Se si deve restituire un valore, lo si salva in x10-x11 (a0-a1).
2. I registri (se salvati) devono essere ripristinati :
 - x9 e x18-x27 (s1-s11)
 - x1 (ra)
 - x8 (fp)
3. SP deve solo aumentare dell'opportuno offset.

Fase 6 - Ritorno al chiamante

Istruzione "jalr x0, 0(x1) (o pseudo-istruzione jr ra).

Fase 7 - Post chiamate lato chiamante

- Eventuale uso del risultato della funzione in x10 e x11 (a0-a1).
- Ripristino dei valori x5-x7 e x28-x31 (t0-t6), x10-x17 (a0-a7) vecchi, se erano stati scritti prima di chiamare la funzione (procedura).

Fasi di una invocazione di una procedura foglia

Le fasi descritte precedentemente si applicano a tutti i tipi di procedura, ma possono essere semplificate, ovvero, se una procedura è foglia (ovvero non chiama a sua volta procedure), nel codice della procedura :

- non è necessario salvare e ripristinare il registro "ra".
- non è necessario salvare sullo stack i registri a0-a7 e t0-t6 (non c'è nessuna invocazione di procedura nella procedura chiamata).

Fase 1 - Pre chiamata del chiamante

- In caso siano usati del chiamante, si devono salvare i registri (solo quelli usati) tra a0-a7 e t0-t6 (potrebbero essere sovrascritti dal chiamato). Vanno salvati nello stack.
- Si devono salvare gli argomenti (o parametri) della funzione. Fino a 8 argomenti possono essere salvati in a0-a7, gli altri vanno salvati nello stack in modo tale che si trovino subito sopra al frame della funzione chiamata.

Fase 2 - invocazione della procedura

Istruzione jal NOME_ISTRUZIONE.

Fase 3 - prologa lato chiamato

Eventuale allocazione del call-frame sullo stack

- salvataggio di x8 (fp), solo se utilizzato all'interno della procedura
- Salvataggio di x9 e x18-x27 (s1-s11) se utilizzati all'interno della procedura (il chiamante si aspetta di trovarli intatti)

Eventuale inizializzazione di fp L punta al nuovo call-frame

Fase 4 - Corpo della procedura

Esecuzione effettiva della procedura.

Fase 5 - Epilogo lato chiamato

- In caso si debba restituire un output, lo si salva in a0/a1.
- In caso si siano utilizzati (e quindi per forza salvati per quanto visto sopra), si ripristinano i registri s1-s11 e fp.

Fase 6 - ritorno al chiamante

Istruzione jalr x0, 0(x1).

Fase 7 - post chiamata lato chiamante

- Eventuale uso del risultato della funzione (salvato in a0/a1).
- Ripristino dei registri a0-a7 o/e t0/t6 se usati prima di chiamare la funzione.

Record di attivazione - struttura

Fp : se utilizzato, viene inizializzato al valore di sp all'inizio della chiamata. Fp permette di avere un riferimento alle variabili locali che non muta con l'esecuzione della procedura.

