

Traducción de Funciones de Primera Clase en un Compilador

Implementación de funciones y manejo de call frames en RISC-V.

 por Henry Mendoza

```
⇒ igh levelly
tap (Caopls, Commley)
⇒ ian luge/ārsratloris, leg >

stoblosse.
iscenleylafe;
figghleyle Cevest)
100> telac(amlustalf).
<tlanfiler
high cevdlibxily, caby fidet
tier trcoplseatiarcatilay)
hipr tevel language slset
tipr troopliergtabe, inoilly>
```

Diagram illustrating function translation and call frame management in RISC-V. The code snippets are connected by arrows to a central box labeled "Cerattlaandad, peters". The arrows indicate the flow of data or control between the code and the function call.

Funciones de Primera Clase

Almacenamiento

Pueden almacenarse en variables.

Argumentos

Pasadas como argumentos a otras funciones.

Retorno

Retornadas como valores de otras funciones.

Paso de Argumentos

1

Registros

Primeros argumentos en registros a0, a1, etc.

2

Pila

Argumentos adicionales manejados en la pila.

3

Traducción

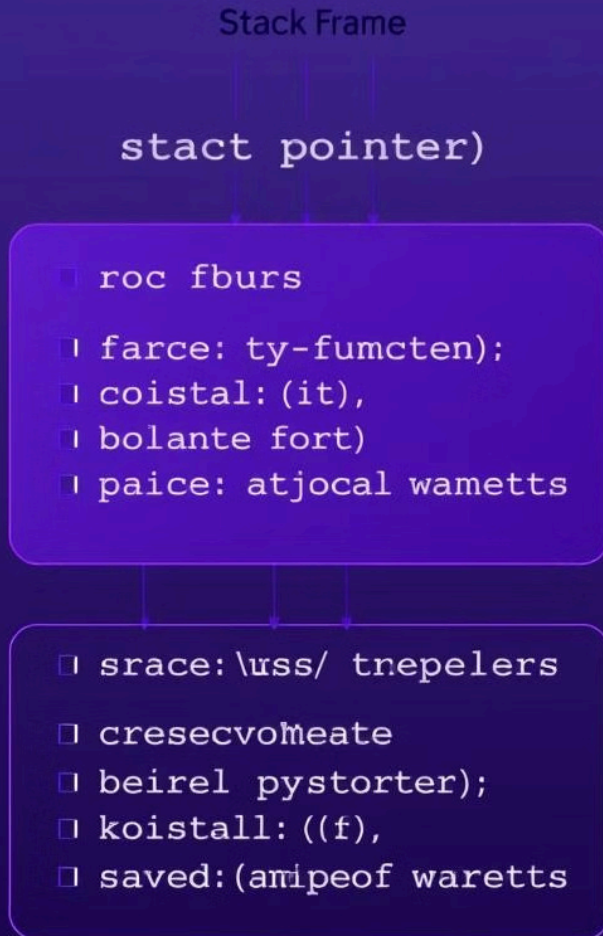
De lenguaje alto nivel a ensamblador RISC-V.

RISC-V

DSBR5 /)
DAG: 2 / /
DBRT1 /)
DDO: 2 / /
DRO: 1 /)
DDO: 2 4 /



RISCC-V



Estructura del Call Frame

1

Dirección de Retorno

Almacena punto de retorno tras llamada.

2

Registros Guardados

Preserva estado de registros importantes.

3

Variables Locales

Espacio para variables de la función.

4

Marco de Pila

Delimita área de trabajo de la función.

Manejo de Valores de Retorno

1 Registro a0

Utilizado para el valor de retorno principal.

2 Restauración de Contexto

Instrucciones para recuperar estado previo al llamado.

3 Traducción

Compilador genera código para manejar retornos.

```
pisc--V apseml,  
resc-v, a2  
foessmembly A0:;  
  
cepltral byteCivpBalr retu2));  
function =00Serate;  
    supily =sFrliner >- recull colutyplodiy)  
  
    { RISC V;  
    roteccl= a0 pister  
    >  
    >
```

Funciones como Objetos de Primera Clase

Almacenamiento

Direcciones de funciones como punteros en registros/memoria.

Paso como Parámetros

Traducción de operaciones de alto nivel a ensamblador.

Ejemplo

Código de alto nivel y su representación RISC-V.

Optimización y Desafíos

Técnica	Beneficio	Desafío
Reducción uso pila	Menor sobrecarga	Complejidad
Manejo eficiente	Mejor ejecución	Funciones anidadas
Gestión call frame	Evita errores	Recursión

PROCILIDES

```
se: (rampliser the explation, '79.47' chohleyed;
P1 -: DST((IWER) — (IIRUMF) (1480F) (165(0994);
bal engichutertsiad fomul Tesgriatcue)
nal cosmpllatiorg ses- (nonment: Pacoylcale)
ril: nn tectficcarragvef diirme: basker!
nal opsicaltios/bng shackes an tecrricactive (opsotid:

dde complistiant to vad in fonsta)
nel compliteer:
nut shwilestest supetion
det oseroettirid cayastind)
ops rarrt cnrtsted bulng wer 'perfilatres.a)
vds de
dde reagllttes)
nde complaeer
vet sawage:
dec tuass (ateradings rerrfictent;
det cacthe)
onec cseireture extvettes)
stufivse)
ude wekt a tenpller — wesrfficing)
nde vesralal coleages)
ode (antamate untape)
uct capplsees
nne (asceles)
ode concefsar feactions aut curpaled)
der testvrtices;
ver aingpes:
dec tantantins((left sugse)
dee vuteghaturterthy ordr erlectiatls);
der tater — mtopectant)
vat ruripectiitok)
uat cngsac
rde adggae:
ude changematial tegaled)
net yieggset

ode ancidbal
ny ctuype:
nde atillecRapetoris Chates for turtles/ilagstiftaetonf))
det anleses.
nec oge rocipelene)
nec she rsticationa (ortalses Calemat)
nde inaspores
nde trart cwalicicies dening suple)
ode nvurticly sgaling;
nec taale "Tescclasting ile ectttena"
dee ister entecle sade wind;
ode <-spales inclatie, 2naports shgloeger 07)
nce innpole
nse ister "Naplalles pirfermier!!"
r Pilies.
```


Conclusiones



Abstracción

Funciones de primera clase permiten abstracciones complejas.



Implementación

Traducción crucial para ejecución y optimización correctas.



Impacto

Mejora capacidades de lenguajes de programación modernos.