OHJ-4200 Laitteistonläheinen ohjelmointi
Tentti 26.5.2010
Pertti Lehtinen

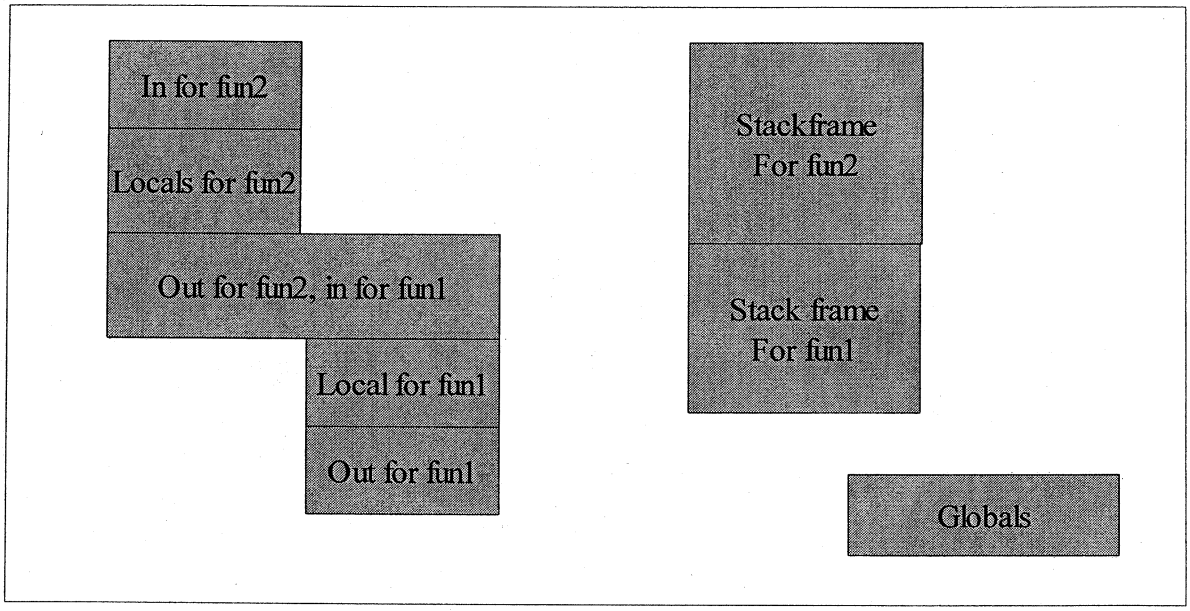
EI LASKINTA!

1. Miten prosessoriarkkitehtuureja luokitellaan operandien määrän avulla.
2. Mitä tarkoittaa osoitusmuoto? Kuvaa ainakin kuusi erilaista.
3. Aktivaatiotietue sisältää aliohjelman suorituksessa tarvittavan informaation. Seuraavassa C++-koodissa suoritus on edennyt merkittyyn kohtaan Merkitse kaavioon eri muuttujien sijainti tuolloin (muuttujat v1,a1,v,x,s1,b,s,a,z,y,t)

```
Vector<int> v1;

Int fun1( int a1, Vector<int> v, int& x, char* s1 )
{
 int b; char s[256];
 //... koodia
**** suorituskohta ****
}

int fun2( int a )
{
  int z,y,char t[256];
  fun1( a, v1, y, t);
  // ... koodia
}
```



Kuvassa SPARC-rekisterit ja pinokehys. Merkitse muuttujien sijainti.

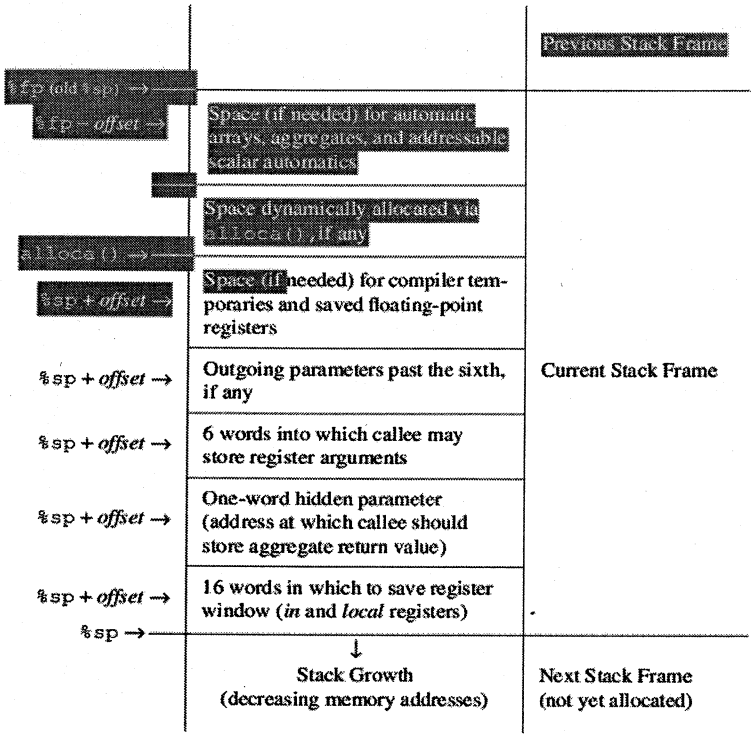


Figure D-2 *The User Stack Frame*

	%i7	(%r31)	return address − 8 †
	%fp, %i6	(%r30)	frame pointer †
	%i5	(%r29)	incoming param 6 †
in	%i4	(%r28)	incoming param 5 †
	%i3	(%r27)	incoming param 4 †
	%i2	(%r26)	incoming param 3 †
	%i1	(%r25)	incoming param 2 †
	%i0	(%r24)	incoming param 1 / return value to caller †
	%l7	(%r23)	local 7 †
	%l6	(%r22)	local 6 †
	%l5	(%r21)	local 5 †
local	%l4	(%r20)	local 4 †
	%l3	(%r19)	local 3 †
	%l2	(%r18)	local 2 †
	%l1	(%r17)	local 1 †
	%l0	(%r16)	local 0 †
	%o7	(%r15)	temporary value / address of CALL instruction ‡
	%sp, %o6	(%r14)	stack pointer †
	%o5	(%r13)	outgoing param 6 ‡
out	%o4	(%r12)	outgoing param 5 ‡
	%o3	(%r11)	outgoing param 4 ‡
	%o2	(%r10)	outgoing param 3 ‡
	%o1	(%r9)	outgoing param 2 ‡
	%o0	(%r8)	outgoing param 1 / return value from callee ‡
	%g7	(%r7)	global 7 (SPARC ABI: use reserved)
	%g6	(%r6)	global 6 (SPARC ABI: use reserved)
	%g5	(%r5)	global 5 (SPARC ABI: use reserved)
global	%g4	(%r4)	global 4 (SPARC ABI: global register variable §)
	%g3	(%r3)	global 3 (SPARC ABI: global register variable §)
	%g2	(%r2)	global 2 (SPARC ABI: global register variable §)
	%g1	(%r1)	temporary value ‡
	%g0	(%r0)	0
state	%y		Y register (used in multiplication/division) ‡
	(*icc* field of %psr)		Integer condition codes ‡
	(*fcc* field of %fsr)		Floating-point condition codes ‡
	(*ccc* field of %csr)		Coprocessor condition codes ‡
	%f31		floating-point value ‡
floating point	:		:
	:		:
	%f0		floating-point value ‡