1. **函数调用栈**

函数调用时，首先反向push形参，然后再push返回地址，即ra寄存器的值。接着将sp的值赋给fp即帧寄存器，并将其push入栈中。再push在外部仍在使用中的需要保护的寄存器。最后是被调用的函数使用当前栈。



图1 函数调用栈示意图

函数返回时，利用fp找到ra寄存器的位置，用一条jr指令完成return。

1. **符号表**

符号表设计采用C99标准，即只有全局符号表和当前的局部符号表，在分支语句中不允许定义变量。访问时首先访问局部符号表，若局部符号表中找不到，则转去全局符号表，若仍找不到，则报错。符号表采用python的字典结构，通过变量名来索引：

全局符号表：

WHOLE\_VALTABLE[idname]={'type':T\_type,'width':4,'value':initVal,'reg':None,'const':ISCONST,'array':True}。

记录了类型、宽度、当前值、当前使用的寄存器、是否为const变量、是否为数组变量这些信息。

局部符号表：

LOCAL\_VALTABLE[idname]={'type':T\_type,'width':4,'offset':str(stack\_offset),'value':initVal,'reg':None,'const':False,'array':True}

记录了类型、宽度、栈中的偏移、当前值、当前使用的寄存器、是否为const变量这些信息。