

Subprogram Implementation Review

COMP3220 – Principle of Programming Languages

2016 Spring

1 Concepts

1. What are the two steps in locating a non-local variable in a *static-scoped* language with stack-dynamic local variables and nested subprograms?
2. What is an EP, and what is its purpose?
3. How are references to variables represented in the static-chain method?

2 Stack Content

Consider the following code snippet.

```
void fun1() {  
    float a_f1;  
}  
  
void fun2() {  
    int a, a_f2, b_f2;  
}  
  
void fun3() {  
    float a_f3, b;  
    // [1]  
}  
  
void main() {  
    char a, a_m, b, b_m;  
}
```

Local Variable
Parameters
Dynamic Link
Static Link
Return Address

Figure 1: Activation Format

The calling sequence is `main` \rightarrow `fun2` \rightarrow `fun1` \rightarrow `fun3`.

The activation record format is in Figure 1.

1. Show the stack content when execution reaches position [1]. Assume that this program uses the *deep-access method* to implement *dynamic scoping* (you may not need the static link field in this case).
2. Assume the program uses the *shallow-access method* (one stack for each variable) to implement *dynamic scoping*. Show the stacks when execution reaches position [1].
3. Assume this program uses *static scoping*, show the stack content when execution reaches position [1].