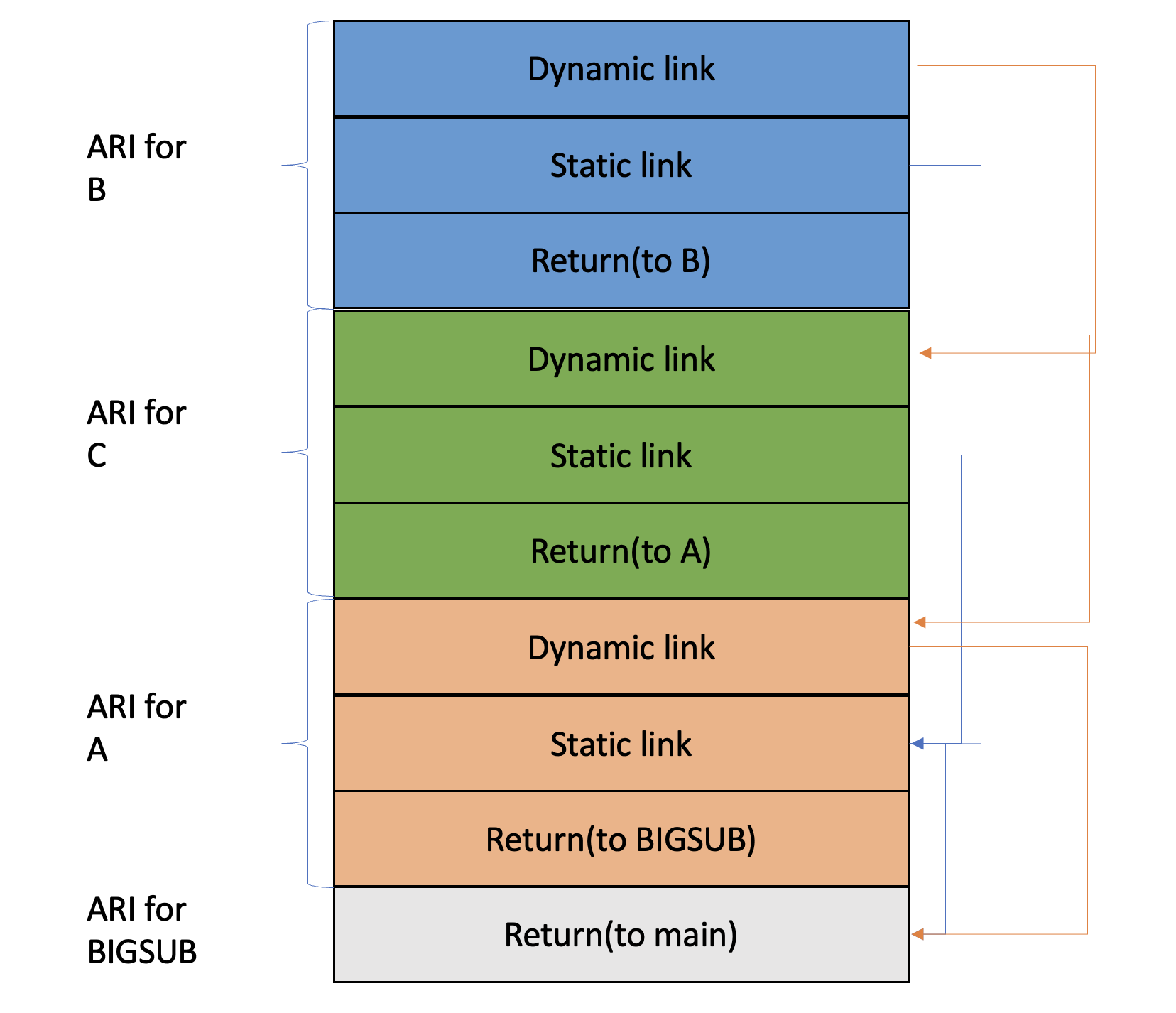
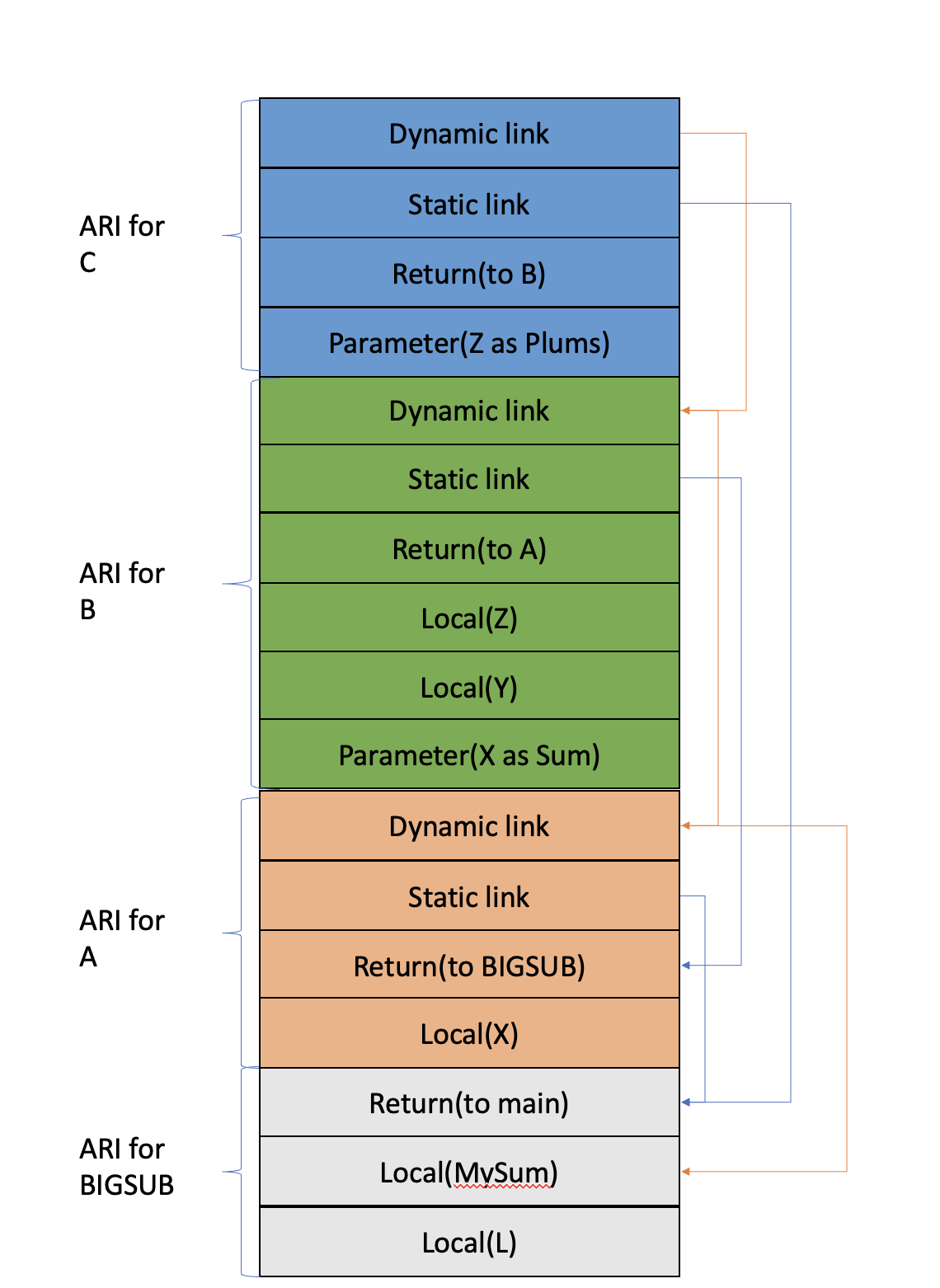
20181662 이건영

<과제 1>

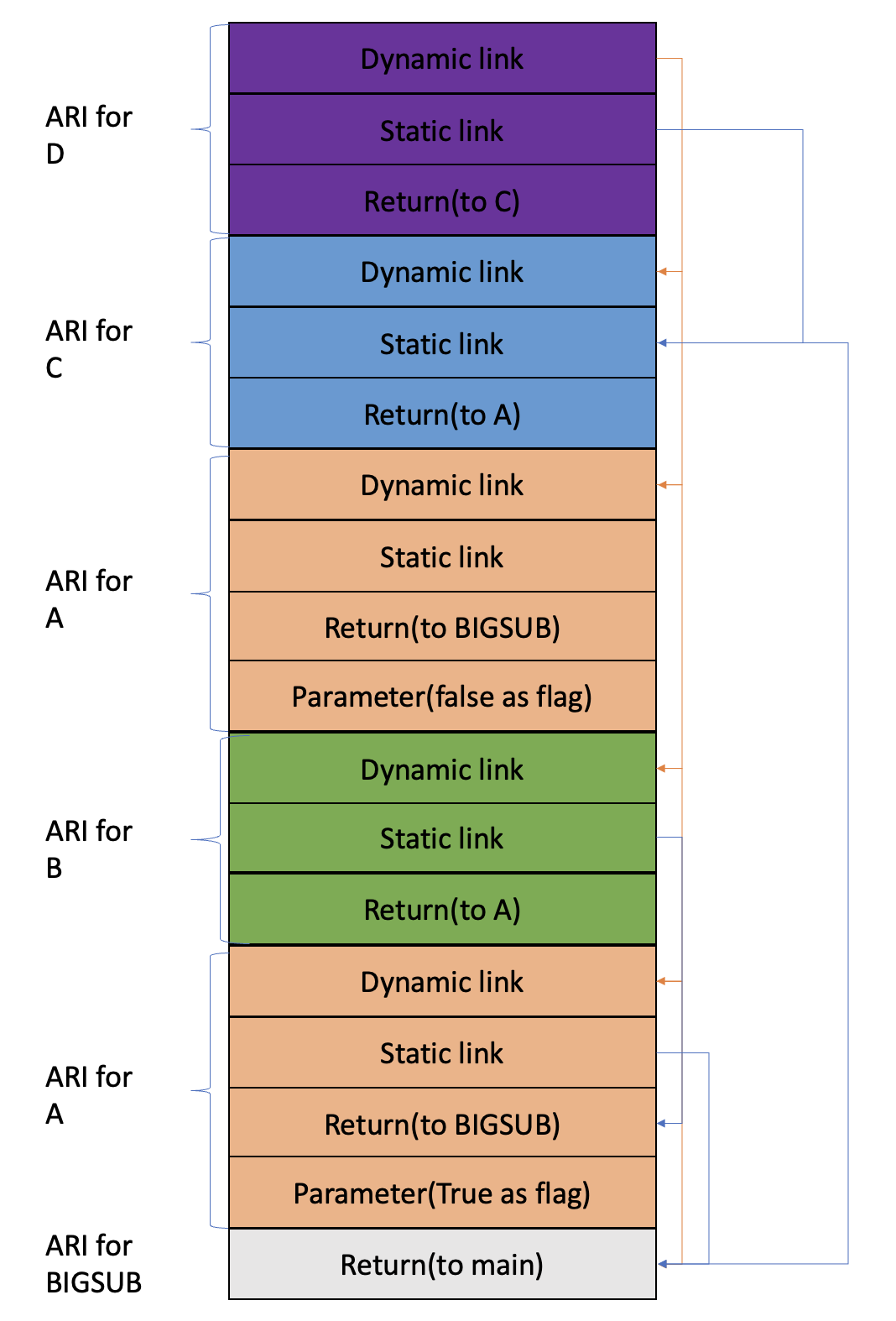
1.



2.



3.



4.

테이블이(가) 표시된 사진

자동 생성된 설명

5.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | fun1 |  |
| main | main | main | fun2 | fun2 | fun1 | fun3 |
| e | f | g | b | c | a | d |

6.

변수들이 static으로 선언되거나 변수를 초기화하는 코드가 수행되지 않는다면 이전과 같은 값을 가진다.

<과제 2>

텍스트이(가) 표시된 사진

자동 생성된 설명

<c파일>

텍스트, 명판이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 화면, 점수판이(가) 표시된 사진

자동 생성된 설명

<s파일>

먼저 s코드의 3번째 줄에서 globalInt 변수를 정의한다. 14번째 줄부터 시작되는 main에서는 지역 변수의 값 할당이 이루어지는데, 26번째 줄부터 32번째 줄이 그에 해당한다. Rbp 기준 -32에는 mainInt가 사이즈 4만큼, -28에 mainSt가 사이즈 8만큼, -20에 mainArr가 사이즈 12만큼 존재한다.

함수 sub1이 호출되기 전 main의 지역변수들을 레지스터에 저장한 후(36~39번째 줄) sub1을 호출한다.

54번째 줄의 sub1로 이동하여 현재의 rbp를 저장한다. 이후 sub1의 종료 후 이 rbp를 이용하여 main으로 return 될 것이다. 63~65번째 줄을 보면 이전에 main에서 저장해둔 값들을 새 rbp의 각 위치에 저장한다. 이후 sub1의 지역변수인 subInt를 -4 위치에 저장한다. 전역변수인 globalInt는 변수의 이름으로 접근하는 것을 확인할 수 있다. 마지막으로 return 값인 10을 보면 10을 eax에 저장시키는 것을 확인할 수 있는데, 이를 39번째 줄에서 -32 위치(mainInt가 저장되었던 위치)에 넣어줌으로써 함수의 return값을 전달하는 것을 확인할 수 있다.