



on Stack-based Linkage Convention

Instructions: For each question, choose the single best answer. Make your choice by clicking on its button. You can change your answers at any time. When the quiz is graded, the correct answers will appear in the box after each question.

1. In the stack-based linkage convention, what subroutines must save the return address to their caller on the stack?

- ☐ A. all subroutines
- ☐ B. only `main`
- ☒ C. any subroutine that calls another subroutine.
- ☐ D. any subroutine that does not call another subroutine.

C

2. Which of the following code fragments saves the return address?

☐ A.

```
sub $ra,$ra,4
sw  $sp,($ra)
```

☒ B.

```
sub $sp,$sp,4
sw  $ra,($sp)
```

☐ C.

```
add $ra,$ra,4
sw  $sp,($ra)
```

☐ D.

```
sw  $ra,($sp)
sub $sp,$sp,4
```

3. When an operating system calls a `main` routine, what is in `$ra`?

- ☐ A. `$ra` is all zeros to indicate the beginning of the call chain.
- ☒ B. The return address to a point in the operating system.
- ☐ C. The entry point to `main`
- ☐ D. Random information.

4. Subroutine A calls subroutine B. Subroutine A has information in `$t0` that it needs preserved. All the `$s` registers are in use. What can be done?

- ☐ A. Subroutine A saves `$t0` in a `$v` register.
- ☐ B. Subroutine B saves `$t0` on the stack.
- ☐ C. Subroutine A saves `$t0` on the stack after calling subroutine B.
- ☒ D. Subroutine A saves `$t0` on the stack before calling subroutine B.

5. What are the two jobs of a subroutine's prolog?

- ☒ A. Save the return address (if needed) and save any `$s` registers it might use.
- ☐ B. Initialize variables and retrieve arguments.
- ☐ C. Save the return address (if needed) and save any `$t` registers it might use.
- ☐ D. Put return values in `$v` registers and restore `$s` registers.

6. What are the jobs of a subroutine's epilog?

- ☐ A. (i) put returned values in `$s` registers, (ii) pop the return address, if needed, (iii) `jx` back to the caller.
- ☐ B. (i) pop the return address, if needed, (ii) `jx` back to the caller.
- ☒ C. (i) put returned values in `$v` registers, (ii) pop any `$s` registers that were pushed, (iii) pop the return address, if needed, (iv) `jx` back to the caller.
- ☐ D. (i) put returned values in `$a` registers, (ii) pop any `$t` registers that were pushed, (iii) pop the return address, if needed, (iv) `jx` back to the caller.

7. In what order are registers popped off the stack when they are being restored?

- ☐ A. The same order that they were pushed.

- ☒ **B.** The reverse of the order that they were pushed.
- ☐ **C.** Alphabetical order.
- ☐ **D.** Numerical order.

8. Is there any limit to how deep subroutine calls can be nested?

- ☐ **A.** Yes. Only `main` can call a subroutine, the the nesting is one level deep.
- ☐ **B.** Yes. Calls can be nested no more than 8 levels deep.
- ☒ **C.** No, as long as there is memory for the stack, calls can be nested any level deep.
- ☐ **D.** No, as long as there is memory on the hard disk to give each subroutine a virtual address space.

9. During execution of an application made up of many subroutines, how does the activation chain behave?

- ☒ **A.** It repeatedly grows and shrinks depending on what subroutines are active.
- ☐ **B.** It constantly grows longer and longer.
- ☐ **C.** It grows longer and develops branches as subroutines call different subroutines.
- ☐ **D.** It grows to its maximum length and then stays there until the program is over.

10. What was the first prominent programming language that used stack-based subroutine linking?

- ☐ **A.** FORTRAN
- ☒ **B.** Algol
- ☐ **C.** COBAL
- ☐ **D.** C

grade quiz

The number you got right:

Percent Correct:

Letter Grade:



If you have returned here from another page, or have re-loaded this page, you will need to click again on each of your choices for the grading program to work correctly. You may want to press the SHIFT KEY while clicking to clear the old answers.

