

## CS300 Exam1 Review

1. What is Linux kernel?
2. What is the shell? Give several shell commands.
3. Linux commands to get around in the file system.
4. scp command
5. difference between scp and ssh
6. pre-processor, compiler, linker, loader
7. uses of and why use #define, static
8. Be able to explain everything in the following makefile ... for example ... what is a target, what is a dependency, how are they used, what is -Wall, what is -g, why use \${CC}, why isn't stk.o a dependency for palindromeChecker.o, ...

```
1 CC=gcc
2 CFLAGS=-Wall -g
3
4 .PHONY: all clean
5
6 all: bin/palindromeChecker
7
8 bin/palindromeChecker: ../StaticStack/bin/stk.o bin/palindromeChecker.o
9   ${CC} ${CFLAGS} bin/palindromeChecker.o \
10      ../StaticStack/bin/stk.o -o bin/palindromeChecker
11
12 bin/palindromeChecker.o: src/palindromeChecker.c \
13      ../StaticStack/include/stk.h
14   ${CC} ${CFLAGS} -c src/palindromeChecker.c -o bin/palindromeChecker.o
15
16 ../CS300StaticStack/bin/stk.o: ../DynamicStack/include/stk.h \
17      ../DynamicStack/src/stk.c
18   cd ../CS300DynamicStack; make bin/stk.o
19
20 clean:
21   rm bin/*.o bin/palindromeChecker
```

9. What is a data structure?
10. What is an ADT?
11. Why use ADTs?
12. Assume the implementation for the String ADT below.

Implement each of the string functions from the String ADT using this representation for a String.

```
typedef struct String
{
    int length;
    char *pszData;
} String;
```

13. Review the Stack ADT and think how else you might want to represent a Stack. There are many. For instance, using a single simple one-dimensional array and no struct.
14. What is the heap? Activation Record (AR)? malloc? free? static versus dynamic memory?
15. Define a struct Person that can hold a name, age, and gender.
16. Define a struct pointer type that can point to a Person struct.
17. Create a pointer to a Person struct and an actual Person variable.
18. Dynamically create memory for a Person and set the pointer to the dynamically allocated memory.
19. null pointer, void pointer, dereference, array of void \*'s, buffer
20. Write an algorithm that determines whether a string of characters is of the form  $x R y$  where  $y$  is the reverse of  $x$ . You can only read one character at a time from the keyboard. So for instance, if the user enters abRba that is an acceptable string while abRab is not.
21. Assume all stack operations from assignment 2 have been implemented. Write the C code to implement your algorithm from 20.
22. Write an algorithm that determines whether a string of characters is of the form  $a R b R c R d R e R \dots$  where  $R$  is defined as in question 20.
23. Assume all stack operations from assignment 2 have been implemented. Write the C code to implement your algorithm from 22.