A18877

THE UNIVERSITY OF BIRMINGHAM

Degree of B.Sc. with Honours
Computer Science/Software Engineering. Final Examination.
Computer Science/Software Engineering with Business Studies. Final Examination.
Artificial Intelligence and Computer Science. Final Examination.
Degree of BEng/MEng with Honours
Computer Science/Software Engineering Final Examination.
Joint Degree of MEng with Honours
Electronic and Software Engineering. Third Examination.

Joint Degree of B.Sc. with Honours
Mathematics and Computer Science. Final Examination.

Degree of MSc in Computer Science.

Occasional Computer Science/Software Engineering

06 17423

Systems Programming in C/C++

May 2006 $1\frac{1}{2}$ hours

[Answer questions 1–3 and one of questions 4 and 5]

Questions 1–3 are compulsory.

1. (a) What is the representation of a string in C? [5%]

   (b) What is the difference between an array and a pointer? [5%]

   (c) The program *linecount* (see Appendix 1) should accept one argument which is a filename. If this file is readable, it should print out each line in turn and at the end the number of lines containing the character "a". Reading stops when an empty line is encountered. Lines which are longer than LINELENGTH characters cause the program to abort. The attached program `linecount.c` tries to do this, but it contains errors. The output from the compiler is

   ```
   gcc -Wall -o linecount linecount.c
   linecount.c: In function 'checkOccurrence':
   linecount.c:38: warning: comparison between pointer and integer
   ```

   Assume the file "test" is

   ```
   Line1a
   Line1b
   LIne1c
   A line with aaaa
   ```

   the output of the program is

   ```
     ./linecount test
   Segmentation fault
   ```

   Checking with the debugger shows that the segmentation fault occurs in line 24. After fixing this error the output of the program is

   ```
   /linecount test
   Having read line Line1a
   Having read line Line1b
   Having read line LIne1c
   Having read line A line with aaaa
   The number of lines containing a is 4
   ```

   with a very long delay after each of the first four lines. The correct output of the program is

   ```
    ./linecountCorrected test
   Having read line Line1a
   Having read line Line1b
   Having read line LIne1c
   Having read line A line with aaaa
   The number of lines containing a is 2
   ```

–3–   THIS IS <u>NOT</u> AN OPEN-BOOK EX-
AMINATION. CANDIDATES MAY
NOT CONSULT ANY REFERENCE
MATERIAL DURING THE SITTING.

NO CALCULATOR

with no delay between any of the lines.

Correct the errors in the program. If your solution contains modification of a line, write down the number of the line you want to modify and the corrected line. If your solution contains addition of a line, write down the number of the line after which you want to add a line of code, and write down the additional line. If your solutions contains deletion of a line, write down the number of the line you want to delete. There are four errors in the program, possibly occurring several times.

[12%]

2. (a) What is a memory leak? [5%]

   (b) What is a buffer overflow? [5%]

   (c) What does "segmentation fault" mean? Which (erroneous) code typically produces it? [5%]

   (d) The program *treeread* (see Appendix 2) reads repeatedly a line of integers and outputs the integers on this line but ordered according to value. Here is an example dialogue for the correct program:

```
 ./treereadCorrected
4 5 3 2
The next element is 2
The next element is 3
The next element is 4
The next element is 5
98 7 2 -1
The next element is -1
The next element is 2
The next element is 7
The next element is 98
```

The compilation output is

```
gcc -g -Wall -o treeread treeread.c
treeread.c: In function 'main':
treeread.c:59: warning: format argument is not a pointer (arg 3)
treeread.c:76: warning: format argument is not a pointer (arg 3)
```

and the result of the program is actually:

```
./treeread
4 5 3 2
Segmentation fault
```

Checking with the debugger shows that the segmentation fault occurs in line 55.

Correct the errors in the program. If your solution contains modification of a line, write down the number of the line you want to modify and the corrected line. If your solution contains addition of a line, write down the number of the line after which you want to add a line of code, and write down the additional line. If your solutions contains deletion of a line, write down the number of the line you want to delete. There are eight errors in the program, possibly ocurring several times and requiring corrections at different places in the program. [20%]

(e) Write a C++-class definition for a tree-structure as given in the file `treeread.c` and write code for the method `display` in the file `treeread.c`. [8%]

3. (a) What is the purpose of a Makefile? [5%]

(b) The enclosed Makefile (see Appendix 3) is supposed to be used for compiling and installing a program called `insert` from the source files `insert.c` and `display.c`. Both files use the include file `trees.h`.

   (i) Is the Makefile suitable for this purpose? If not, correct it. [5%]

   (ii) Which commands are executed by running the corrected Makefile with the listing here?

```
-rwxr--r--  1 exr staff  318 Feb  7 17:37 display.c
-rw-r--r--  1 exr staff  960 Feb  7 17:38 display.o
-rwxr--r--  1 exr staff  943 Feb  7 17:37 insert.c
-rw-r--r--  1 exr staff 1488 Feb  7 17:38 insert.o
-rwxr-xr-x  1 exr staff  573 Feb  7 17:38 Makefile
-rwxr--r--  1 exr staff  184 Feb  7 17:44 trees.h
```

[5%]

   (iii) Is this Makefile suitable for creating an RPM? If not, indicate which modifications are necessary. [5%]

Answer one of the following two questions.

4. (a) Describe the hazards of kernel programming. [5%]

(b) How does data exchange between the kernel and the user program work? [5%]

(c) Why is it important even on a monoprocessor system to ensure that shared data structures in the kernel are modified in critical sections? [5%]

5. (a) What is the purpose of templates in C++? [5%]

(b) How do you implement the equivalent of Java-interfaces in C++? [5%]

(c) How do you pass functions as parameters in C/C++? [5%]

A18877