NO CALCULATOR PERMITTED IN THIS EXAMINATION

## THE UNIVERSITY OF BIRMINGHAM

## THIS PAGE TO BE REPLACED BY OFFICE

06 15258

# Operating Systems with C/C++

May 2014  $1\frac{1}{2}$  hours

[Answer ALL questions]

A28171

1. (a) Describe the possible states of a process and the transitions between them.

#### [6%]

- (b) List two pre-requisites for scheduling algorithms to be successful and justify their necessity. [6%]
- (c) Consider a shop selling online videos. Orders are processed by a dedicated server. Customers can either obtain a preview of parts of the video, or can buy the whole video. Generating the previews is computationally expensive, whereas buying the videos is I/O-intensive but not computationally expensive. Describe a suitable scheduling strategy for such a scenario. Your scheduling strategy should maximise throughput. You should also explain your reasons for choosing this strategy. [8%]
- 2. (a) Describe the working set model for allocating pages to processes. **[6%]** 
  - (b) A laboratory is used for teaching first-year programming as well as running CPU-intensive processes for research in the background. Before upgrading the memory-intensive IDE which is used for programming, the CPU was normally in use for nearly 100% of the time, and the response time was short. After the upgrade the CPU is still in use for nearly 100% of the time, but the response time has increased dramatically, and throughput has been drastically reduced. Give a possible cause for this behaviour and suggest a remedy which does not involve additional hardware.
  - (c) Data received from the hard disk is usually written into kernel buffers using DMA without CPU-interference. Is it possible to write such data directly into the memory area of the user process which receives the data? Justify your answer. [6%]

[5%]

- 3. (a) What is a pointer in C/C++? How can it be used to write a function which returns several values? [5%]
  - (b) How are strings represented in C?
  - (c) Consider the following code fragment:

```
1
     struct List {
 2
        int element;
 3
        struct List *next;
 4
      };
 5
 6
   struct List *list = NULL, *new, *last;
 7
 8
   int i = 3;
 9
10 while (i >0 ) {
      new->next = NULL;
11
12
      new->element = i;
13
      last->next = list;
14
      list = new;
15
      i--;
16
     }
```

After the execution of this code fragment the variable <code>list</code> should point to a list with the elements 3, 2 and 1, and the variable <code>last</code> should point to the last element of the list. The given code does not satisfy these properties. Identify the errors in the code and suggest corrections. [10%]

[6%]

4. (a) Describe access control lists.

## (b) The ptrace-system call makes it possible to observe and control the execution of some process and change its memory and register. Is it safe to allow the ptrace-system call for a process which results from the execution of a program with the suid-bit set? Justify your answer. [7%]

- (c) Assume any process can access any part of the memory. In this case is it possible for the operating system to ensure that a user cannot gain access to the password of anyone else? Justify your answer. [7%]
- 5. (a) What is the effect of a segmentation fault in a kernel program? [5%]
  - (b) What is a system call? How is it used in a program? [5%]
  - (c) Consider the following kernel code fragment:

```
#define BUFFERSIZE 256;
1
2
   char data[BUFFERSIZE];
З
4
  int readData (struct file *file, const char *buffer,
5
                        unsigned long count, void *data) {
6
       memcpy (data, buffer, count);
7
       processData (data, count);
8
       return count;
    }
9
```

Assume that the procedure readData is called whenever a user process writes to a certain file in the proc-directory. More precisely, the variable buffer points to the data written by the user process, and the variable count indicates the size of this data. Note that the variable buffer points to user space. The procedure readData should return the number of bytes read or -1 if there was an error.

This kernel code fragment is supposed to call the function processData for the data written by the user process. It compiles correctly, but has several errors concerning at least memory management and handling of concurrency. List the errors and correct them. For critical sections it is sufficient to indicate begin and end of a critical section, and whether you would use semaphores or spinlocks to protect the critical section. **[10%]**