Digital Systems: Problem sheet 4

Mike Spivey, Hilary Term, 2022

1 In micro:bian, what happens if the function that forms the body of a process returns? What happens if all such functions return?

2 In micro:bian, what happens if a process tries to send a message to itself?

3 A micro:bian process can have a list of processes waiting to send to it. Imagine a directed graph in which the nodes are processes, and there is an arrow to each process from all the other processes that are waiting to send to it. What happens if there is a cycle in this graph? How could such cycles be detected?

4 The program shown in Figure 1 contains two processes that share a variable `r`: one process increments `r` from 0 to 100,000 while the other prints the value of `r` ten times. What might we see as output from this program? Would it make a difference if the calls to `start` in `init` were re-ordered?

```
#include "microbian.h"

static volatile int r = 0;

void proc1(int n) {
    for (int i = 0; i < 10; i++)
        printf("r = %d\n", r);
}

void proc2(int n) {
    while (r < 100000) r++;
}

void init(void) {
    serial_init();
    start("Proc1", proc1, 0, STACK);
    start("Proc2", proc2, 0, STACK);
}
```

Figure 1: *Program for Exercise 4*


```
#include "microbian.h"

void put_string(char *s) {
    for (char *p = s; *p != '\0'; p++)
        serial_putc(*p);
}

static const char *slogan[] = {
    "no deal is better than a bad deal\n",
    "BREXIT MEANS BREXIT!\n"
};

void speaker(int n) {
    while (1)
        put_string(slogan[n]);
}

void init(void) {
    serial_init();
    start("May", speaker, 0, STACK);
    start("Farage", speaker, 1, STACK);
}
```

Figure 2: *Program for Exercise 6*

5 Consider a situation where a process is continuously sending characters to the serial driver. The processor time for a typical context switch to send and receive a message is about 20 μsec.

(a) How many context switches happen for each character sent?

(b) How much can the UART speed be increased before context switching time occupies a substantial fraction of the time that the UART takes to send a character?

(c) Suggest a way of reducing the number of context switches per character output.

6 The program shown in Figure 2 was written to display political slogans, but (un)fortunately its output is garbled. Why? Closer examination reveals that characters from the two slogans alternate in the output: "nBoR EdXeIaTl MiEsA NbSe. . . ". Why does that happen?

Design a modification to the program that (unlike the Today programme on Radio 4) allows each speaker to complete a sentence before the other one intervenes. If your first solution involves the two speakers transmitting their slogans via a coordinating process (the 'presenter'), design another solution where the presenter does not handle the text of each slogan, but only coordinates them by giving them permission to speak, one at a time.

(For authenticity, the two speakers in this simulation repeat the same, tired phrases over and over again, but your solution should also accommodate a more fruitful debate, where the two speakers concoct a series of new lies, using some method that cannot be delegated to the presenter.)

7 micro:bian provides an operation

```
sendrec(dest, type, &m);
```

that is equivalent to the two calls,

```
send(dest, type, &m);
receive(REPLY, &m);
```

It is useful as a form of 'remote procedure call', where a client process sends a request to a server process and then waits for a reply. Outline, in terms of process states, how this operation can be implemented. What efficiency advantages does it offer, compared with the equivalent `send` followed by `receive`? How does using `sendrec` help to ensure process priorities are respected in a situation where a low-priority client sends a request to a high-priority server process?

8 The interface of `receive` requires that a process be prepared to accept a message either of a specific type, or any message at all. Suggest changes to the interface and the implementation of micro:bian that would allow any set of acceptable message types to be specified.