

1. (20) К Написати и објасните *Fine grain Ticket* алгоритам реализован помоћу *FA* операције. Уколико би уместо *FA* постојала операција *Compare-And-Swap* која би недељиво обављала (

CAS(a, b, c) :

```
< if (a == c)
    { c = b; return true;}
else { a = c; return false;}>;
```

да ли је могуће направити *Fine grain* решење и ако је могуће – направите га.

2. (20) К Посматра се проблем читалаца и писаца (*Readers-Writers Problem*). Користећи семафоре написати код за писце и за иницијализацију уколико је код за читаоце дат. Семафор *rw* се иницијализује на почетну вредност N ($N > 1$), а поред овог семафора читаоци не користе другу синхронизацију.

procedure Reader(id : integer);

begin

```
wait(rw);
read_db;
signal(rw)
```

end;

3. (20) Реализовати филтерску мрежу за елиминацију истих вредности из непразног низа целих бројева који се завршава са *EOS*. Процеси имају до две локације за смештање елемената низа. Поруке које се размењују могу да садрже само по један елемент низа. Користити асинхрону размену порука. Сваки процес треба да има идентичан код. Приказати филтерску мрежу која најбрже обрађује 19 улазних вредности. Коментарисати уређење елемента резултујућег низа.

4. (20) Користећи библиотеку *C-Linda* решити проблем забавишта (*The Child Care Problem*). У неком забавишту постоји правило које каже да се на свака три детета мора наћи барем једна васпитачица. Родитељ доводи једно или више деце у забавиште. Уколико има места оставља их, уколико не, чека. Васпитачица сме да напусти забавиште само уколико то не нарушава правило. Написати процедуре за родитеље који доводе и одводе децу и васпитачице и иницијализовати почетне услове.

Испитиј итраје 3 сати.

Напомена: На вежбанци назначити задатке које не треба прегледати.

1. (20) К Написати и објасните *Fine grain Ticket* алгоритам реализован помоћу *FA* операције. Уколико би уместо *FA* постојала операција *Compare-And-Swap* која би недељиво обављала (

CAS(a, b, c) :

```
< if (a == c)
    { c = b; return true;}
else { a = c; return false;}>;
```

да ли је могуће направити *Fine grain* решење и ако је могуће – направите га.

2. (20) К Посматра се проблем читалаца и писаца (*Readers-Writers Problem*). Користећи семафоре написати код за писце и за иницијализацију уколико је код за читаоце дат. Семафор *rw* се иницијализује на почетну вредност N ($N > 1$), а поред овог семафора читаоци не користе другу синхронизацију.

procedure Reader(id : integer);

begin

```
wait(rw);
read_db;
signal(rw)
```

end;

3. (20) Реализовати филтерску мрежу за елиминацију истих вредности из непразног низа целих бројева који се завршава са *EOS*. Процеси имају до две локације за смештање елемената низа. Поруке које се размењују могу да садрже само по један елемент низа. Користити асинхрону размену порука. Сваки процес треба да има идентичан код. Приказати филтерску мрежу која најбрже обрађује 19 улазних вредности. Коментарисати уређење елемента резултујућег низа.

4. (20) Користећи библиотеку *C-Linda* решити проблем забавишта (*The Child Care Problem*). У неком забавишту постоји правило које каже да се на свака три детета мора наћи барем једна васпитачица. Родитељ доводи једно или више деце у забавиште. Уколико има места оставља их, уколико не, чека. Васпитачица сме да напусти забавиште само уколико то не нарушава правило. Написати процедуре за родитеље који доводе и одводе децу и васпитачице и иницијализовати почетне услове.

Испитиј итраје 3 сати.

Напомена: На вежбанци назначити задатке које не треба прегледати.