Esercizio E2.1

Impostazione

1)Quali processi?

Esistono tre tipi di processi: quelli di tipo M che leggono e scrivono file multimediali, quelli di tipo B che effettuono il backup e quelli di tipo R che ripristinano i dati di backup di processi.

2)Quale struttura per i processi?

Per ogni processo la struttura è la seguente:

```
cliente
   storage.accedi
   <esegui il compito>
   storage.rilascia
```

dove storage rapresenta una particolare istanza del monitor **storage_system**

Soluzione

```
# define  N… ; /*numero di workstation */
typedef enum{M, B, R}tipo; /*tipo di cliente*/
typedef enum{true,false}boolean;
typedef int workstation;
```


/*operazioni del monitor */

```
monitor storage_system {
   /*variabili del monitor: */
   /* num. di processi per tipo che stanno usando lo storage: */
   int numero[3];
   /* per ogni workstation, dice se c'è già un processo R o B: */
   boolean occupato[N];
   /* code su cui sospendere i processi: */
   condition coda[3];
   /*numero di processi in coda per tipo: */
   int incoda[3];
```

/* Procedure entry*/

```
   public void accedi(tipo t, workstation w) {
      if (t == M)
```

```
      /* multimedia: non ci devono essere backup */
         while (numero[B] > 0)
            {incoda[M]++; wait (coda[M]); incoda[M]--; }
      else if (t == R)
      /*restore: non ci deve essere il corrispondente backup */
         while (occupato[w])
            {incoda[R]++; wait (coda[R]); incoda[R]--; }
      else /*caso t = B */
      /* backup: non c'è multimedia né il corrispondente restore */
         while ((numero[M] > 0) || (occupato[w]))
            {incoda[B]++; wait (coda[B]); incoda[B]--; }
      numero[t]++;
      if ((t== R) || (t == B))
         occupato[w] = true;
   } /* fine accedi*/

   procedure rilascia(tipo t, workstation w) {
      int s;
      if ((t== R) || (t == B))
         occupato[w] = false;
      numero[t]--;
      if ((t== M) && (numero[M]== 0)) {
         s = incoda[B];
         for (i = 0; i<s; i++)
            signal(coda[B]);
      }
      if (t== B) {
         if (numero[B]== 0) {
            s = incoda[M];
            for (i = 0; i<s; i++)
                   signal(coda[M]);
         }
         s = incoda[R]; /* risveglia tutti, ma al massimo uno esegue */
         for (i = 0; i<s; i++)
            signal(coda[R]);
      }
      if (t==R) {   /* risveglia tutti,, ma al massimo uno esegue */
         s = incoda[B];
         for (i = 0; i< s;i++)
            signal(coda[B]);
      }
   }/* fine rilascia*/

}/*fine monitor*/
```