Università degli Studi di Napoli Federico II
Corso di Laurea in Ingegneria Informatica
Esame di Sistemi Operativi
Proff. Cinque, Cotroneo, Natella

Prova pratica del 17/05/2021
Durata della prova: 75 minuti

Lo studente completi il programma a corredo di questo documento, seguendo le seguenti indicazioni.

La prova sarà valutata come segue:

- **A**: Prova svolta correttamente.
- **B**: Il programma non esegue correttamente, con errori minori di programmazione o di concorrenza.
- **C**: Il programma non esegue correttamente, con errori significativi (voto max: 22).
- **INSUFFICIENTE**: Il programma non compila o non esegue, con errori gravi di sincronizzazione.

Istruzioni per la consegna dell'elaborato

L'elaborato dovrà essere svolto in una cartella dal nome: `Cognome_Nome_Matricola_Docente`

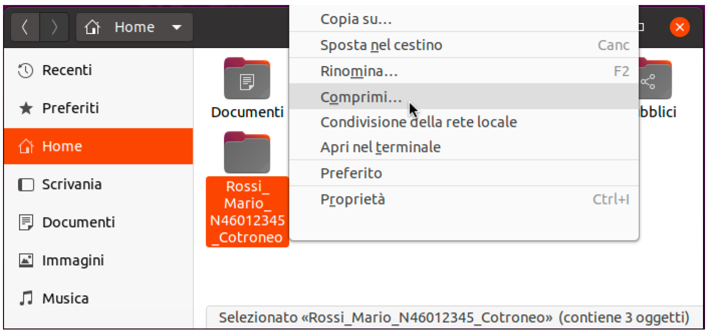
Esempio:

```
mkdir Rossi_Mario_N46012345_Cotroneo
cd Rossi_Mario_N46012345_Cotroneo
```

…. copiare nella cartella i file forniti per l'esame ….
…. sviluppare il programma ….
…. per la consegna, dalla shell (assumendo di essere ancora nella cartella di lavoro), creare un file compresso ("tar") con i seguenti comandi:

```
cd ..
tar -czvf ./Rossi_Mario_N46012345_Cotroneo.tar.gz  ./Rossi_Mario_N46012345_Cotroneo
```

In alternativa, è consentito creare il file compresso "tar" tramite l'interfaccia grafica.



All'interno della macchina virtuale, aprire il browser all'indirizzo:

Nel form, cliccare su "**aggiungi file**", selezionare il file compresso contenente il proprio svolgimento, e indicare il proprio nome ed email @studenti.unina.it.

Attendere una notifica del docente, e quindi scollegarsi dalla piattaforma di VirtualClassroom.

Testo della prova

Si realizzi in linguaggio C/C++ un programma **multithread**, basato sul **costrutto Monitor**, che realizzi lo schema del **produttore-consumatore con vettore di stato**. È previsto che i produttori possano inserire elementi di due tipi differenti. I buffer del vettore possono essere in 4 possibili stati: LIBERO, IN_USO, OCCUPATO1, OCCUPATO2 (questi ultimi due stati a seconda del tipo di elemento che è stato prodotto). I produttori chiamano uno fra due differenti metodi ("produci_tipo_1" e "produci_tipo_2") per effettuare la produzione sui due tipi differenti. Analogamente, i consumatori chiamano uno fra due metodi differenti ("consuma_tipo_1" e "consuma_tipo_2") per consumare.

Per verificare il funzionamento del programma, si creino 2 thread produttori, che effettuino 4 produzioni ciascuno di tipo 1; altri 2 thread produttori, che effettuino 4 produzioni ciascuno di tipo 2; 1 thread consumatore, che effettui 8 consumazioni di tipo 1; 1 thread consumatore, che effettui 8 consumazioni di tipo 1. Il vettore di buffer conterrà al più 4 elementi.

```
#define DIM 4

typedef struct {
     int vettore[DIM];
     int stato[DIM];
     int num_liberi;
     int num_occupati_tipo1;
     int num_occupati_tipo2;

     // ... aggiungere ulteriori variabili per la sincronizzazione
} MonitorPC;

void inizializza(MonitorPC * m);
void rimuovi(MonitorPC * m);
void produci_tipo_1(MonitorPC * m, int valore);
void produci_tipo_2(MonitorPC * m, int valore);
void consuma_tipo_1(MonitorPC * m, int & valore);
void consuma_tipo_2(MonitorPC * m, int & valore);
```