

**Studente:** {{Nome}} {{Cognome}}, Matr. {{Matricola}}, Email: {{Email}}

**Compito:** {{Num}} **Turno:** {{Dsati}}

## PRINCIPI DI SISTEMI OPERATIVI

A.A. 2015/16

22 Gennaio 2016

### IMPORTANTE:

- Si considerano parte integrante delle soluzioni i COMMENTI significativi introdotti per facilitare la lettura del codice: come tali, essi influenzano la votazione finale. Tuttavia, i messaggi di debug (ad es. le `println()`) del programma NON SONO CONSIDERATI E QUINDI NON INFLUENZANO LA VOTAZIONE FINALE.
- Il tempo a disposizione è di 2 ore.
- Il compito deve essere svolto in Java, lavorando con l'ambiente di sviluppo Eclipse, usando le classi del package `monitor` o i costrutti di sincronizzazione nativi di Java.
- Seguire le seguenti regole per lo svolgimento dell'esame::
  - a. Fare il login in Linux con il proprio account
  - b. Aprire un browser web, andare su `ftp://lica02.lab.unimo.it`
  - c. Seguire le istruzioni del file README per la creazione della directory **studente\_XXXX** sul Desktop.
  - d. Aprire Eclipse ed impostare la directory **studente\_XXXX** come workspace.
  - e. Creare un progetto Java chiamandolo **studente\_NumeroMatricola**.
  - f. Finito l'esame: salvare, chiudere Eclipse, fare il logout, lasciare il vostro PC e procedere alla consegna del testo.

### TESTO:

Una compagnia marittima dispone di una **imbarcazione** ed offre tour panoramici di una baia ai viaggiatori. Ciascun tour parte da un **porto**, dura un certo lasso di tempo e torna indietro. L'imbarcazione dispone di **N** posti. La compagnia accetta sia viaggiatori **singoli** che in **gruppo** (**n** persone con  $2 \leq n \leq N$ ). I viaggiatori in gruppo hanno priorità rispetto ai viaggiatori singoli. L'imbarcazione rimane in attesa di caricare per un certo periodo di tempo, dopodichè salpa ed effettua il tour (di durata costante). Al termine del tour i passeggeri sbarcano e se ne vanno.

Si implementi una soluzione che modelli i viaggiatori e l'imbarcazione come Thread ed il porto come risorsa, assicurandosi che la gestione delle code sia sempre FIFO e garantendo la priorità indicata. Nella soluzione si massimizzi l'utilizzo delle risorse. Si discuta se la soluzione proposta può presentare starvation e in caso positivo per quali processi, e si propongano modifiche e/o aggiunte per evitare la starvation (nello spazio bianco qui sotto).

*"A good traveller has no fixed plan, and is not intent on arriving" (Lao Tzu)*