Secondo esercizio d'esame: problema dei selvaggi

- Una tribù di N selvaggi mangia in comune da una pentola che può contenere fino ad M porzioni di stufato, si assume che inizialmente la pentola sia piena. Quando un selvaggio ha fame controlla la pentola: se non ci sono porzioni, sveglia il cuoco ed attende che questo abbia riempito di nuovo la pentola; se la pentola contiene almeno una porzione, se ne appropria. Il cuoco controlla che ci siano delle porzioni e, se ci sono, si addormenta, altrimenti cuoce M porzioni e le mette nella pentola.
- Utilizzare le system call POSIX per la gestione di semafori (sem_init, sem_wait e sem_post) e le funzioni della libreria pthread per la gestione dei thread

Problema dei selvaggi

- Il comportamento del selvaggio e del cuoco senza sincronizzazione è il seguente:

Cuoco:
Loop
    - Riempi Pentola
End

Selvaggio:
Loop
    - Pensa
    - Prendi porzione da Pentola
    - Mangia
end

- I vincoli di sincronizzazione sono:
  - I selvaggi non possono servirsi se la pentola è vuota
  - Il cuoco può riempire la pentola solo se è vuota
- La soluzione usa le primitive classiche dei semafori, quindi escludendo:
  - molteplici operazioni su semafori atomiche (System V)
  - letture del valore del semaforo (System V)

- Le variabili da usare sono:
  - Intero Porzioni: tiene traccia del numero di porzioni ancora in pentola
  - Semaforo Mutex: per accedere alla variabile Porzioni
  - Semaforo Vuoto: indica pentola vuota
  - Semaforo Pieno: indica pentola piena

Suggerimenti

```
Cuoco()
{
    while(1){
        down(vuoto)
        <riempi pentola>
        up(pieno)
    }
}
```

```
Selvaggio()
{
    for(i=0;i<NGIRI; i++){
        <pensa>
        …
        if(porzioni==0) {
            …
        }
        porzioni--
        …
        <mangia porzione>
    }
}
```