USER LEVEL THREADS and FIRING SQUAD SYNCHRONIZATION PROBLEM

Assignment 2
OS142
BGU
User level Threads

• Thread data structure
  – Stack (ESP, EBP): every thread must maintain it’s *private* stack
  – Registers: *local* to current thread execution

• Stack manipulations
  – Thread creation: allocate thread data, create initial stack frame
  – Context switch: update current stack (assumes corresponding stack frame), update register values
User level Threads [initialization]

• Allocate a stack for a new thread (one page size: 4096 bytes)

• Create initial stack frames:
  – Inner frame corresponds to thread execution function (return address points to uthread_exit())
  – Outer frame must be a context switch function frame
Stack conventions

For more information you can use: http://www.cs.virginia.edu/~evans/cs216/guides/x86.html
FSS Model

- Finite one-dimensional cellular array consisting of $n$ cells, denoted by $C_i$, where $1 \leq i \leq n$.
- All cells (except the end cells) are identical finite state automata.
- The array operates in lock-step mode such that the next state of each cell (except the end cells) is determined by both its own present state and the present states of its right and left neighbors.
- All cells, except the left end cell, are initially in the quiescent state at time $t = 0$ and have the property whereby the next state of a quiescent cell having quiescent neighbors is the quiescent state.
- At time $t = 0$ the left end cell (general) is in the fire when ready state, which is an initiation signal to the array.
FSS PROBLEM

• Given an array of \( n \) identical cellular automata, including a *general* on the left end which is activated at time \( t = 0 \), we want to give the description (state set and next-state function) of the automata so that, *at some future time*, all of the cells will *simultaneously* and, *for the first time*, enter a special *firing* state.

• The set of states must be independent of \( n \). Without loss of generality, we assume \( n \geq 2 \).
Task 4

• Implement platform for FSS Model
  – This is where you use your synchronization primitives (micro synchronization)

• Implement algorithm for FSSP
  – Macro synchronization
Algorithms
# Algorithms

[Table and diagram content]
Algorithms