CS-5600 - Computer Systems

Lecture 8

Midterm Review
- I/O & DMA
- Disks, block devices, RAID

Context Switching

Main
- Call getline
- Call putline

getline
- Call putchar
- Loop
- If c = \n
putchar
- Getchar
- Call yield
- Loop

Monitors

A
- putchar
- put
- m2
- m3
- A + m1 + m2 + m3

3
```
count = 0
sum = 0
condition C

m1(=m2=m3)(x)
  count++
  sum += x
  if count <= 3
    wait C
  else
    broadcast C
  endif
  return sum/3
  count = 0
  average = sum/3
  sum = 0
  return average
```

Full solution

```
bool m1_busy
condition C1

m1:
  if m1_busy
    wait (C1)
    m1_busy = True
  else
    signal C1
```

SIO and DMA

enable NOT enable

Data lines (8/16/32 bits)

Address lines

Frame buffer doesn't (may not) read all address lines

FrameBuffer

IEEE memory suspend

Good RAM

~ 36 MHz

1 cycle

1000 cycles

Device

DDR2/DDR3

RAM: bus

L1: L2: L3 cache - internal to chip

DMA - Direct Memory Access

Ask device to pump data into a particular area
Standard way of doing DMA.

With 1 CPU & 1 I/O device:

CPU - insert commands into buffer/queue
I/O device - removes work items from buffer

Memory

I/O register

DMA descriptors

CPU

- prepare DMA descriptors, describe the CPU
- put descriptors (pointers) in circular buffer
- update 'head' ptr of circular buffer (I/O unit)
Device
- read_packet of descriptor
- use the addr to read descriptor
- read data & do something or read data to
  do some operation like writing data to something
- set status

\( \text{interrupt (I/W interrupt CPU)} \)

\( \text{CPU} \)

\( \text{signal} \) \( \rightarrow \text{read status} \)

done

\( \text{CPU} \) \( \rightarrow \text{device} \)

\( \text{do something} \)

\( \text{device} \)

\( \text{DMAMUX} \) also refers to I/O devices able to access memory directly

Timeline

\( \text{user} \)

\( \text{read()} \)

\( \text{kernel} \)

\( \text{wait()} \)

\( \text{signal() raised by Int-handler} \)

\( \text{hardware} \)

\( \text{Int} \)