CG2271 Real-Time Operating Systems

Tutorial 8 Suggested Solutions

Q1. Suppose the holes available for allocation are of the following size in that order:
10KB, 4KB, 20KB, 18KB, 7KB, 9KB, 12KB, 15KB

Three processes A, B, and C with the respective sizes of 12 KB, 10 KB and 9 KB are to be allocated successively. Describe the results of the allocation when the following allocation methods are used:

a. first fit
b. best fit
c. worst fit
d. next fit

Which algorithm makes use of the memory space the best?

Answer:

First Fit
12KB allocated out of 20KB: 8KB left
10KB allocated out of 10KB
9KB allocated out of 18KB: 9KB left

Best Fit:
12KB allocated out of 12KB
10KB allocated out of 10KB
9KB allocated out of 9KB
This is the best allocation

Worst Fit:
12KB allocated out of 20KB: 8KB left
10KB allocated out of 18KB: 8KB left
9KB allocated out of 15KB: 6KB

Next Fit:
12KB allocated out of 20KB: 8KB left
10KB allocated out of 18KB: 8KB left
9KB allocated out of 9KB

Q2. Consider the following code fragment:

```
int x;
int *p;
int f=0;
int y = 0;

main(){
   int g;

   p = malloc(100);
   f = 6;
   sub(f, g);
}

void sub(int a, int b){
  int c;

 c = a;
 a = x;
 b = y;
 x = y;
 *p = b;
}
```

Explain where each of the variable will be located in memory? Draw the content of the stack memory when sub ( ) is executing.

Answer:

x and pointer p are in BSS
f and y are in data segment
g is allocated on stack
array of 100 elements allocated on heap
a and b are paramaters that are put on the stack
c is allocated on stack

Top of stack
c : function sub
b
a
g : function main
Bottom of stack

Q3. In this question we will consider a virtual memory system with 64 bytes per page, 10 bit virtual addresses and 9 bit physical addresses.

a. What is the maximum size in bytes of the virtual memory? Physical memory?

b. What is the maximum number of virtual and physical pages?

c. Using the page table below, translate the following virtual addresses to physical addresses: 0, 15, 576, 987, 1020

	V	PPN
0	1	2
1	1	3
2	0	
3	0	
4	1	4
5	1	7
6	0	
7	0	
8	0	
9	1	6
10	0	
11	0	
12	1	5
13	0	
14	0	
15	1	1

Answer:

a. 1024 bytes of VM, 512 bytes of physical memory.

b. Byte index is 6 bits (2^6=64), leaving 4 bits for VPN or 16 virtual pages
3 bits for PPN or 8 physical pages.

VA	Binary (virtual page number -- Byte Index)	VPN	PPN	PA Binary (physical page no- Byte Index)	PA
0	0000 000000	0	2	0010 000000	128
15	0000 001111	0	2	0010 001111	143
576	1001 000000	9	6	0110 000000	384
987	1111 011011	15	1	0001 011011	91
1020	1111 111100	15	1	0001 111100	124

Q4. You are required to develop a program that will move 1MByte of data from address 0xC0000000 to 0xD0000000. You must update the progress of the data transfer for every 1Kbyte of data. This is done by updating an LCD panel that is connected to the system.

Discuss two ways in which the objective can be realized. Elaborate on their pros and cons.

Answer:

The primitive approach would be to use a loop to access each memory location and transfer the data one word/byte at a time. For every multiple of 1Kbyte, we can update the LCD panel. A sample pseudo-code will look something like this:

```
for(i = 0; I < 1Mbyte; i++)
{
  *(dest_address + i) = *(source_address + i);
   If(I % 1024 == 0)
      Update LCD Screen
}
```

The more efficient approach would be to use DMA to perform the data transfer and allow interrupts to update the LCD panel. The set-up would be as follows:

I. Write the start address and number of words/bytes (1Kbytes) to be transferred into the DMAC's registers.

II. Set the DMAC controller to automatically increment the address after each transfer.

III. Configure the DMAC to generate an Interrupt after the transfer is complete.

IV. With each Interrupt, update the LCD and initiate the next DMA transfer.

The advantage of the first approach would be that it can be applied to any microprocessor as it does not depend on any specific hardware. For the second approach, it is definitely more efficient in that it does not use CPU cycles for the data transfer. However, this approach would require the processor to have the necessary DMA hardware blocks, and it may not be available in low-end processors.

THE END