



## Operating Systems HW#7

Due date: 11/27/2002 (Wed)

### Performance Comparison: First-fit vs. Best-fit

가 internal fragmentation, external fragmentation, allocation failure ratio

#### Conditions:

- : 16 Kbytes ( $2^{14}$ bytes)
- : 10, 50, 100 bytes (file names: "mem\_size\_10.dat", "mem\_size\_50.dat", mem\_size\_100.dat")
  - ( uniform distribution 1 random number , 100 bytes , 1~199 random , 1~199 50.25 Random number 가 .)
- : 10 time unit (file name: "alloc\_time\_10.dat")
  - (uniform distribution 1 random number .)
- : 50 time unit (file name: "dealloc\_time\_50.dat")
  - (uniform distribution 1 random number .)

#### Measuring method:

- Simulation , hw07.out .
- Memory allocation policy: First-fit or Best-fit
  - Internal fragmentation: ( ) / ( )
    - ( internal fragmentation .)
  - External fragmentation: ( ) / ( )
    - (External fragmentation , .)
  - Allocation failure ratio: ( ) / ( )

#### What you submit:

- In diskette: hw07.c, hw07.exe, hw07.out (가 window .)
- In hardcopy: hw07.c, hw07.out

#### Appendix A. Description of first-fit memory allocation policy

: doubly linked list .

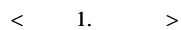
:

```
struct free_block {
```

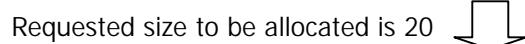


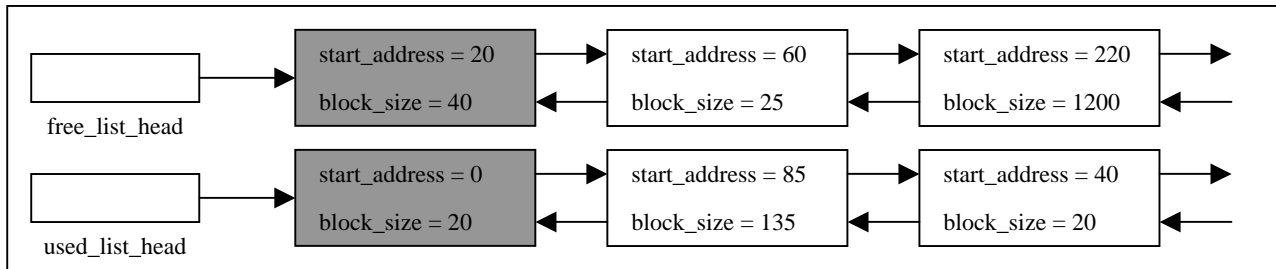
• • • • • 0

1 가 .



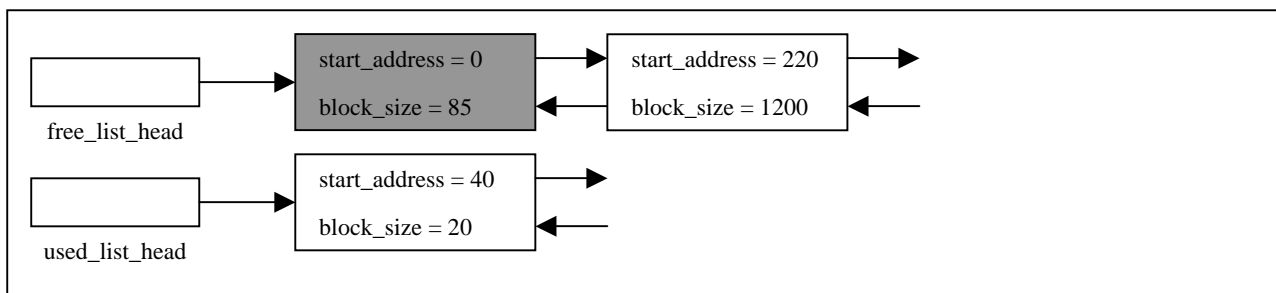
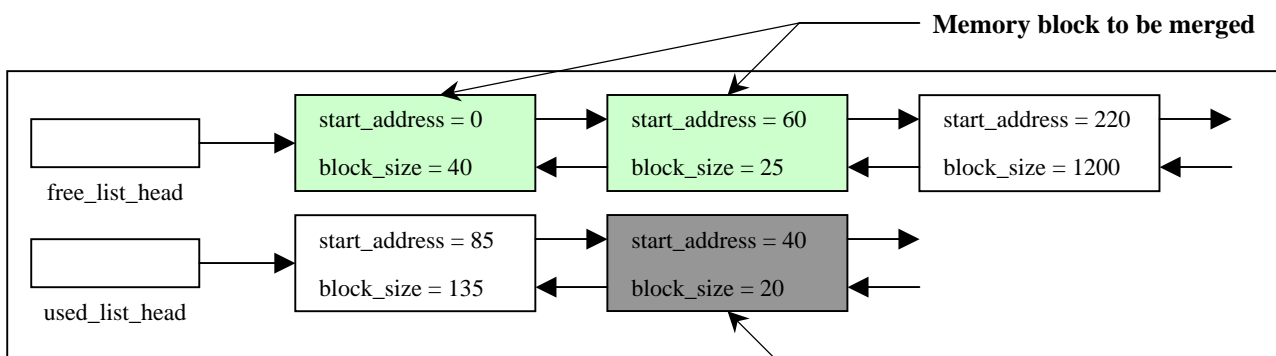
RefBlockNumber ++





< 2. >

: block\_number ( )  
( ) (start\_address )

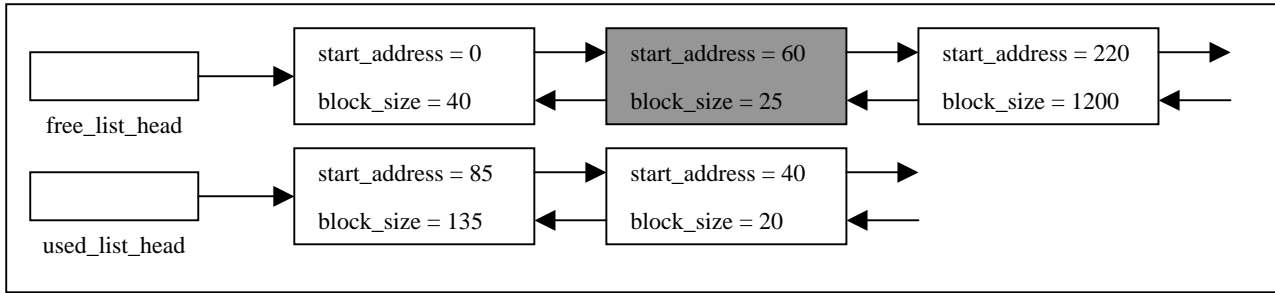


< 3. >

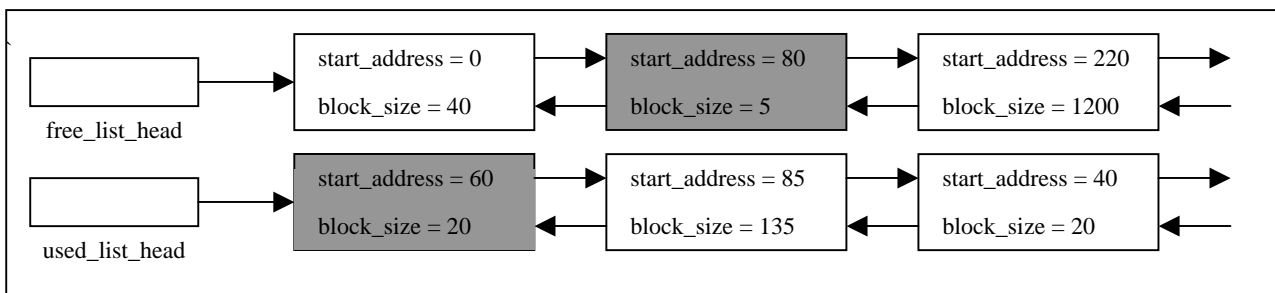
#### Appendix B. Description of best-fit memory allocation policy

First-fit

가  
가  
가  
(start\_address )  
block\_number block\_number  
= RefBlockNumber ++



Requested size to be allocated is 20



< 4. >

#### Appendix C. Code fragment

```
/* The name of this simulation code is hw07.c.

Simulation result should be stored in file hw07.out. */

#define NUM_ALLOCATION 10000          /* The number of memory allocation (or deallocation) */
#define TOTAL_MEMORY_SIZE 16384     /* 2^14 */

long int AllocSize[NUM_ALLOCATION];   /* Allocation Size */
long int AllocTime[NUM_ALLOCATION];   /* Allocation Time */
long int DeallocTime[NUM_ALLOCATION]; /* Deallocation Time */
long int RefBlockNumber, TotalSimulTime;
long int AllocFailCount, AllocReqCount; /* The number of allocation failure, The number of allocation request */
double InterFrag, SumOfExterFrag;    /* Internal Fragmentation, Sum of External Fragmentation */
long int TotAllocSize, TotReqSize;    /* Total Size of Allocated Memory Block, Total Size of Requested Memory Block (Internal frag.) */
long int CurAllocSize;               /* Current Size of Allocated Memory Blocks (External frag.) */
...
main()
{
    input_data ();
    init_global_variables();
    simulation_first_fit();
```



# The School of Information and Communication Engineering Sungkyunkwan University

```
init_global_variables();

simulation_best_fit();

print_simul_result();

}

input_data() {

    /* Fill in these arrays AllocSize[], AllocTime[], and DeallocTime[] by reading files "mem_size*.dat", "alloc_time_10.dat", and
    "dealloc_time_50.dat", respectively. */

    DeallocTime[0] += AllocTime[0];

    for (i = 1; i < NUM_ALLOCATION; i++) {

        AllocTime[i] += AllocTime[i-1];

        DeallocTime[i] += AllocTime[i];

    }

    TotalSimulTime = DeallocTime[i-1];    /* Total simulation time */

}

init_global_variables() {

    RefBlockNumber = 0;

    AllocFailCount = AllocReqCount = 0;

    InterFrag = SumOfExterFrag = 0.0;

    TotAllocSize = TotReqSize = CurAllocSize = 0;

    ...

}

simulation_first_fit() {

    int alloc_index=0, dealloc_index, sp_dealloc=0;    /* index for using memory allocation and deallocation */

    int alloc_block_size, dealloc_block_size, vtime;    /* size of allocated block, size of deallocation block, and virtual time */

    for (vtime = 0; vtime <= TotalSimulTime; vtime++) {

        /* Memory Allocation */

        if (AllocTime[alloc_index] == vtime) {

            AllocReqCount++;

            if ( (alloc_block_size = allocation_first_fit( AllocSize[alloc_index] )) == -1 ) {    /* If allocation request is failed... */

                AllocFailCount++;

                SumOfExterFrag += ( TOTAL_MEMORY_SIZE / CurAllocSize );

            }

            else {    /* If allocation request is succeed... */

                TotReqSize += AllocSize[alloc_index];

            }

        }

    }

}
```



# The School of Information and Communication Engineering Sungkyunkwan University

```
TotAllocSize += alloc_block_size;

CurAllocSize += alloc_block_size;

}

alloc_index++;

}

/* Memory Deallocation */

dealloc_index = sp_dealloc; /* sp_dealloc is start point of searching for DeallocTime[] (to find memory block to be deallocated) */
while (dealloc_index <= alloc_index) {

    if (DeallocTime[dealloc_index] == vtime) {

        if ((dealloc_block_size = deallocation_first_fit( dealloc_index )) == -1) { /* If deallocation is failed... */

            printf("Error..");

            exit(1);

        }

        CurAllocSize -= dealloc_block_size;

        if (sp_dealloc == dealloc_index)

            sp_dealloc++;

    }

    dealloc_index++;

}

...

}

InterFrag = ((double)TotAllocSize) / TotReqSize;

}

int allocation_first_fit( int req_size ) {

    /* return (-1), when allocation request is failed

    return (size of allocated memory block), when allocation request is succeed */

}

int deallocation_first_fit( int blk_num ) { /* blk_num is equal to the block_number field of the block to be deallocated */

    /* return (-1), when deallocation request is failed

    return (size of deallocated memory block), when deallocation request is succeed */

}

print_simul_result( ) {

    printf("\nResult of First Fit Memory Allocation\n");

    printf("Internal Fragmentation: %f\n", InterFrag);

    printf("External Fragmentation: %f\n", (SumOfExterFrag / AllocFailCount));

    printf("Allocation Failure Ratio: %f\n", (((double)AllocFailCount) / AllocReqCount));

}
```