

The background of the slide features a photograph of a large, classical-style statue of a woman in a long, flowing dress, standing with her arms outstretched. The statue is set in a park-like environment with trees and other figures in the background. The entire image is overlaid with a semi-transparent orange color. The text is white and positioned in the upper left and center of the slide.

Fragmentation, Threads, and pthreads

CS 240 - The University of Illinois

Wade Fagen-Ulmschneider

September 9, 2021

Fragmentation

The image features a central photograph of a statue on a pedestal, with the words 'ALMA MATER' visible on the base. The statue is surrounded by a crowd of people, and the background consists of bare tree branches. The entire image is overlaid with a semi-transparent orange filter. The word 'Fragmentation' is written in large, white, bold, sans-serif font across the center of the image.

Fragmentation

Internal Fragmentation:

External Fragmentation:

Fragmentation Example

Unallocated (3072 bytes)
Used (1024 bytes)
Used (1024 bytes)
Used (2048 bytes)
Free (1024 bytes)
Free (2048 bytes)
Used Data (2048 bytes)

Start of Heap



Threads

A photograph of a statue on a pedestal with the inscription "ALMA MATER" and "1863" below it, surrounded by a crowd of people. The entire image is overlaid with a semi-transparent orange filter.

Threads

Threads

A _____ is an organization of one or more threads in the same context.

A simple process has only one thread.

Threads

In C, the initial thread is called the

_____.



pthread

```
int pthread_create(  
    pthread_t *thread,          /* thread struct */  
    const pthread_attr_t *attr, /* usually NULL */  
    void *(*start_routine) (void *), /* start func */  
    void *arg                    /* thread start arg */  
);
```

A photograph of a crowd gathered around a statue of Alma Mater at a university, overlaid with a semi-transparent orange filter. The statue is the central focus, standing on a pedestal. The crowd consists of many people, mostly seen from the back or side, looking towards the statue. The background shows trees and a building. The text 'fifteen-threads.c' is overlaid in white, bold, sans-serif font across the middle of the image.

fifteen-threads.c

fifteen-threads.c

```
1 #include <stdio.h>
2 #include <pthread.h>
3 #include <stdlib.h>
4
5 const int num_threads = 15;
6
7 void *thread_start(void *ptr) {
8     int id = *((int *)ptr);
9     printf("Thread %d running...\n", id);
10    return NULL;
11 }
12
13 int main(int argc, char *argv[]) {
14     // Create threads:
15     int i;
16     pthread_t tid[num_threads];
17     for (i = 0; i < num_threads; i++) {
18         pthread_create(&tid[i], NULL, thread_start, (void *)&i);
19     }
20
21     printf("Done!\n");
22     return 0;
23 }
24
```

fifteen-threads.c

```
1 #include <stdio.h>
2 #include <pthread.h>
3 #include <stdlib.h>
4
5 const int num_threads = 15;
6
7 void *thread_start(void *ptr) {
8     int id = *((int *)ptr);
9     printf("Thread %d running...\n", id);
10    return NULL;
11 }
12
13 int main(int argc, char *argv[]) {
14     // Create threads:
15     int i;
16     pthread_t tid[num_threads];
17     for (i = 0; i < num_threads; i++) {
18         pthread_create(&tid[i], NULL, thread_start, (void *)&i);
19     }
20
21     printf("Done!\n");
22     return 0;
23 }
24
```

fifteen-threads.c

```
1 #include <stdio.h>
2 #include <pthread.h>
3 #include <stdlib.h>
4
5 const int num_threads = 15;
6
7 void *thread_start(void *ptr) {
8     int id = *((int *)ptr);
9     printf("Thread %d running...\n", id);
10    return NULL;
11 }
12
13 int main(int argc, char *argv[]) {
14     // Create threads:
15     int i;
16     pthread_t tid[num_threads];
17     for (i = 0; i < num_threads; i++) {
18         pthread_create(&tid[i], NULL, thread_start, (void *)&i);
19     }
20
21     printf("Done!\n");
22     return 0;
23 }
24
```

A photograph of a statue of Alma Mater, likely from a university, surrounded by a crowd of people. The image is overlaid with a semi-transparent orange filter. The text "fifteen-join.c" is centered in white.

fifteen-join.c

fifteen-join.c

```
13 int main(int argc, char *argv[]) {
14     // Create threads:
15     int i;
16     pthread_t tid[num_threads];
17     for (i = 0; i < num_threads; i++) {
18         int *val = malloc(sizeof(int));
19         *val = i;
20         pthread_create(&tid[i], NULL, thread_start, (void *)val);
21     }
22
23     // Joining Threads
24     for (i = 0; i < num_threads; i++) {
25         pthread_join(tid[i], NULL);
26     }
27
28     printf("Done!\n");
29     return 0;
30 }
```

fifteen-join.c

```
13 int main(int argc, char *argv[]) {
14     // Create threads:
15     int i;
16     pthread_t tid[num_threads];
17     for (i = 0; i < num_threads; i++) {
18         int *val = malloc(sizeof(int));
19         *val = i;
20         pthread_create(&tid[i], NULL, thread_start, (void *)val);
21     }
22
23     // Joining Threads
24     for (i = 0; i < num_threads; i++) {
25         pthread_join(tid[i], NULL);
26     }
27
28     printf("Done!\n");
29     return 0;
30 }
```

fifteen-join.c

```
13 int main(int argc, char *argv[]) {
14     // Create threads:
15     int i;
16     pthread_t tid[num_threads];
17     for (i = 0; i < num_threads; i++) {
18         int *val = malloc(sizeof(int));
19         *val = i;
20         pthread_create(&tid[i], NULL, thread_start, (void *)val);
21     }
22
23     // Joining Threads
24     for (i = 0; i < num_threads; i++) {
25         pthread_join(tid[i], NULL);
26     }
27
28     printf("Done!\n");
29     return 0;
30 }
```