

Socket program

From the pthread_join manual page

The pthread_join() function shall suspend execution of the calling thread until the target thread terminates, unless the target thread has already terminated.

Code snippet (Sequential)

```
for(i=0; i<3; i++){
```

```
    sin_size=sizeof(struct sockaddr_in);
```

```
    if((new_fd=accept(sockfd,(struct sockaddr *)&clientaddr,
```

```
        &sin_size))<0) {          {
```

```
        perror("Oops.\n");
```

```
}
```

```
pthread_create(&tid, &attr, thread, (void *)new_fd);
```

```
    pthread_join(tid, NULL);
```

```
}
```

Can operate in parallel

```
for(i=0; i<3; i++){  
  
    sin_size=sizeof(struct sockaddr_in);  
    if((new_fd=accept(sockfd,(struct sockaddr *)&clientaddr,  
        &sin_size))<0) {  
        perror("Oops.\n");  
    }  
  
    pthread_create(&tid[i], &attr, thread, (void *)new_fd);  
}  
for (j=0; <3; j++) {  
    pthread_join(tid, NULL);  
}
```

Cannot operate in parallel

```
for(i=0; i<3; i++){  
  
    sin_size=sizeof(struct sockaddr_in);  
    if((new_fd=accept(sockfd,(struct sockaddr *)&clientaddr,  
        &sin_size))<0) {  
        perror("Oops.\n");  
    }  
  
    pthread_create(&tid[i], &attr, thread, (void *)&new_fd);  
}  
for (j=0; <3; j++) {  
    pthread_join(tid, NULL);  
}
```

```
void * rthread(void *arg)
{ sleep(4); /* we likely won't get the right value
here as passed by address */
    int new_fd=(int)arg;
    int nbytes;
sem_wait(&mutex);
/* do receive, put in buffer and sleep if you want.
I passed in the received buffer, myself as a unique
address */
sem_post(&mutex);
```

The Internet

- Material on the internet belongs to those who create it!
- If you use it you MUST reference it.
- If used in code, that part of the grade goes to the people who created it.
- Never use other material without citation! It can end a career.