Pthread (9A)

- Pthread

Young Won Lim
01/19/2013

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Pthread Creation and Termination

```
int pthread_create (   pthread_t * thread,
                       const pthread_attr_t * attr,
                       void *(*start_routine) (void*),

                       void * arg                    );
```

stores the **ID** of the created thread
in the location referenced by **thread**

```
void pthread_join(pthread_t thread, void **status);
```

to **wait** for a thread to **terminate**

```
void pthread_detach(pthread_t thread);
```

an alternative to pthread_join() to
reclaim storage for a thread

```
void pthread_exit(void *value_ptr);

int pthread_cancel(pthread_t thread);

int pthread_attr_init(pthread_attr_t *attr);

int pthread_attr_destroy(pthread_attr_t *attr);
```

Creating a Default Thread

```
int pthread_create (   pthread_t * thread,
                       const pthread_attr_t * attr,
                       void *(*start_routine) (void*),
                       void * arg                  );
```

```
pthread_t tid;
pthread_attr_t tattr;
extern void *start_routine(void *arg);
void *arg;
int ret;
```

```
ret = pthread_create(&tid, NULL, start_routine, arg);
```

Create default attributes

```
ret = pthread_attr_init(&tattr);
ret = pthread_create(&tid, tattr, start_routine, arg);
```

Default Attributes

scope	PTHREAD_SCOPE_PROCESS	: unbounded
detachstate	PTHREAD_CREATE_JOINABLE	: nondetatched
stackaddr	NULL	: default stack
stacksize	1 megabyte	: default stack size
inheritsched	PTHREAD_INHERIT_SCHED	: inherit parent's priority

Waiting for a thread to terminate

```
void pthread_join(pthread_t thread, void **status);
```

to **wait** for a thread to **terminate**

works only for target threads that are nondetached

blocks the calling thread
until the specified thread **terminates**.

```
pthread_t tid;
pthread_attr_t tattr;
extern void *start_routine(void *arg);
void *arg;
int ret;
```

```
ret = pthread_create(&tid, NULL, start_routine, arg);
```

```
ret = pthread_join(&tid, NULL);
```

```
ret = pthread_attr_init(&tattr);
ret = pthread_create(&tid, tattr, start_routine, arg);
```

exit code of the defunct thread

```
int status;
ret = pthread_join(&tid, NULL, status);
```

Detaching a thread

```
void pthread_detach(pthread_t thread);
```

an alternative to pthread_join() to reclaim storage for a thread that is created with a detachstate attribute set to PTHREAD_CREATE_JOINABLE.

If tid has not terminated, pthread_detach() does not cause it to terminate.

```
pthread_t tid;
pthread_attr_t tattr;
extern void *start_routine(void *arg);
void *arg;
int ret;
```

```
ret = pthread_create(&tid, NULL, start_routine, arg);
```

```
ret = pthread_attr_init(&tattr);
ret = pthread_create(&tid, tattr, start_routine, arg);
```

```
ret = pthread_join(&tid, NULL);
```

exit code of the defunct thread

```
int status;
ret = pthread_join(&tid, NULL, status);
```

```
ret = pthread_detach(&tid);
```

Terminate a thread

```
void pthread_exit(void *value_ptr);

int pthread_cancel(pthread_t thread);
```

- By returning from its first (outermost) procedure, the threads start routine;
- By calling **pthread_exit**(), supplying an exit status
- By termination with **pthread_cancel**()

```
pthread_t tid;
pthread_attr_t tattr;
extern void *start_routine(void *arg);
void *arg;
int ret;
```

```
ret = pthread_create(&tid, NULL, start_routine, arg);
```

```
ret = pthread_attr_init(&tattr);
ret = pthread_create(&tid, tattr, start_routine, arg);
```

```
ret = pthread_join(&tid, NULL);
```

exit code of the defunct thread

```
int status;
ret = pthread_join(&tid, NULL, status);
```

```
int status;
ret = pthread_exit(&status);
```

```
ret = pthread_cancel(tid);
```

Reference

References

[1] http://en.wikipedia.org/
[2] http://www.tldp.org/LDP/lpg/node46.html