```
//Rut tien
//Compile: gcc -o vd vd.c -lpthread
#include <pthread.h>
#include <stdio.h>
int taikhoan = 900;

/* This is our thread function.  It is like main(), but for a thread*/
void *threadFunc(void *arg)
{
	int tienrut = (int) arg;
	if (taikhoan - tienrut > 0)
	{
		sleep(1);
		taikhoan = taikhoan - tienrut;
		printf("taikhoan = %d \n", taikhoan);
	}
	else
		printf ("Khong the rut tien!\n");
	return NULL;
}

int main(void)
{
	pthread_t pth1, pth2;    // this is our thread identifier
	int tienrut1 = 600;
```

```
	int tienrut2 = 400;
	pthread_create(&pth1,NULL,threadFunc,(void*)tienrut1);
	pthread_create(&pth2,NULL,threadFunc,(void*)tienrut2);
	pthread_join(pth1,NULL);
	pthread_join(pth2,NULL);
	return 0;
}
```

```
//Rut_tien Semaphore
//Compile: gcc -o vd vd.c -lpthread
#include <pthread.h>
#include <stdio.h>
#include <semaphore.h>
sem_t mutex;
int taikhoan = 900;
/* This is our thread function.  It is like main(), but for a thread*/
void *threadFunc(void *arg)
{
	int tienrut = (int) arg;
	sem_wait(&mutex);
	if (taikhoan - tienrut > 0)
	{
		sleep(1);
		taikhoan = taikhoan - tienrut;
		printf("taikhoan = %d \n", taikhoan);
	}
	else
		printf ("Khong the rut tien!\n");
	sem_post(&mutex);
	return NULL;
}
int main(void)
{
```

```
	sem_init(&mutex, 0, 1);
	pthread_t pth1, pth2;    // this is our thread identifier
	int tienrut1 = 600;
	int tienrut2 = 400;
	pthread_create(&pth1,NULL,threadFunc,(void*)tienrut1);
	pthread_create(&pth2,NULL,threadFunc,(void*)tienrut2);
	pthread_join(pth1,NULL);
	pthread_join(pth2,NULL);
	sem_destroy(&mutex);
	return 0;
}
```