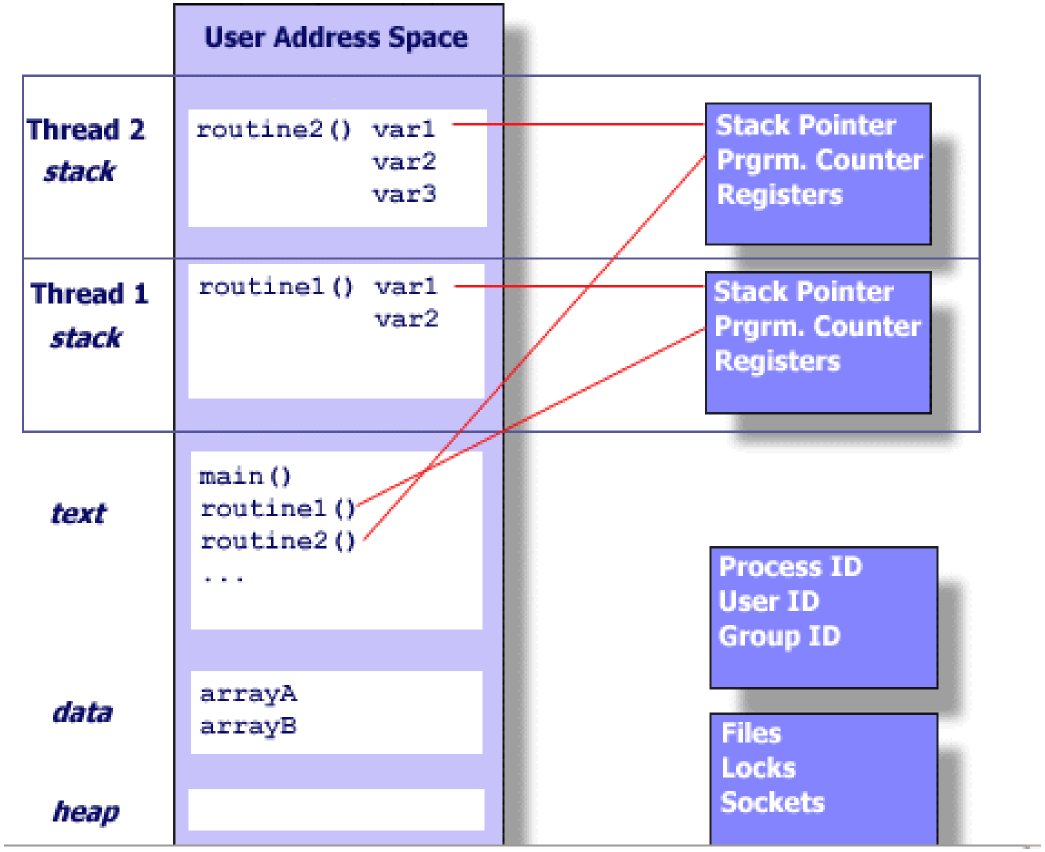
Project hy335b
Threads examples
by Alexmil

7/3/2016

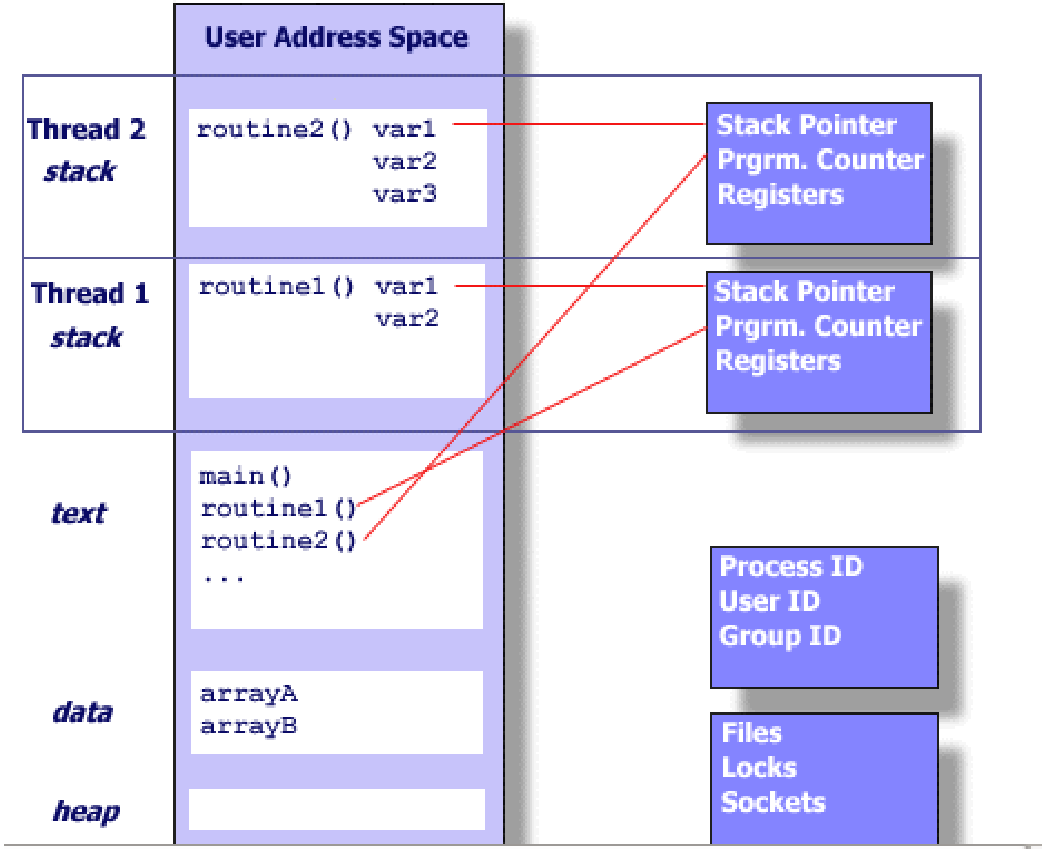
What is a thread(1/3)

- A "procedure" that runs independently from its main program



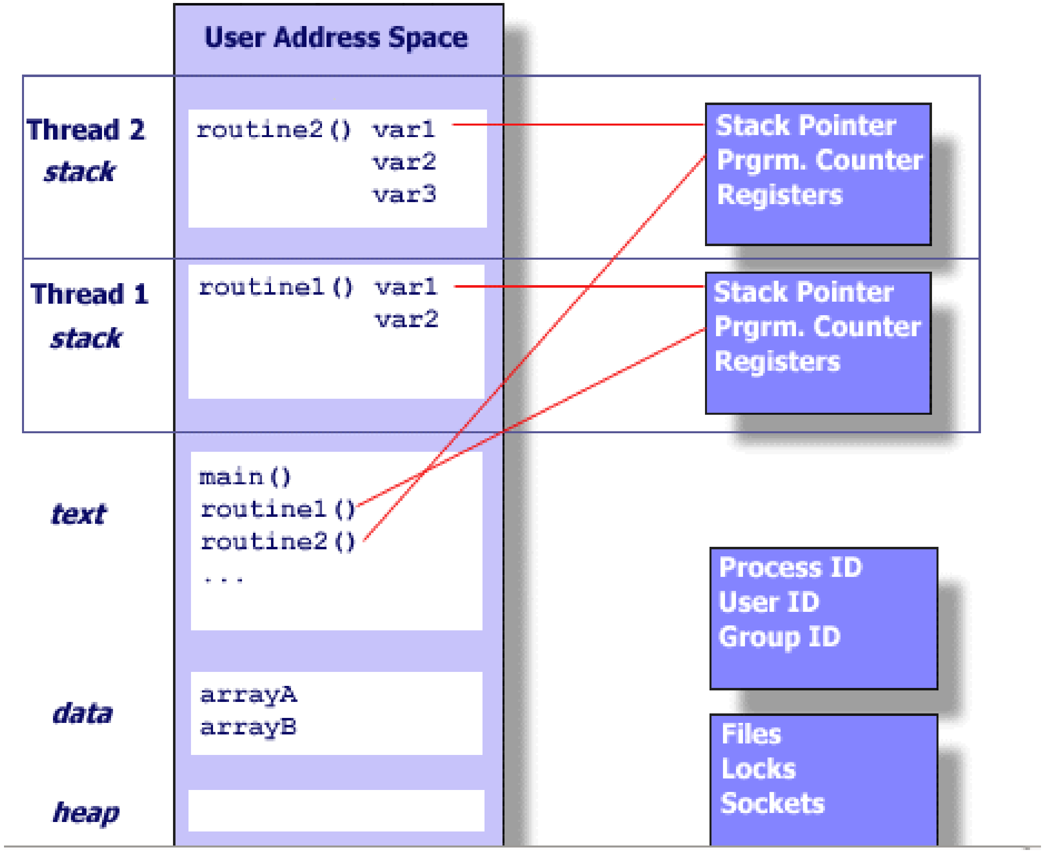
- Can be considered as lightweight processes
- Threads use and exist within these process resources, yet are able to be scheduled by the operating system and run as independent entities because they duplicate only the bare essential resources that enable them to exist as executable code.

What is a thread(2/3)



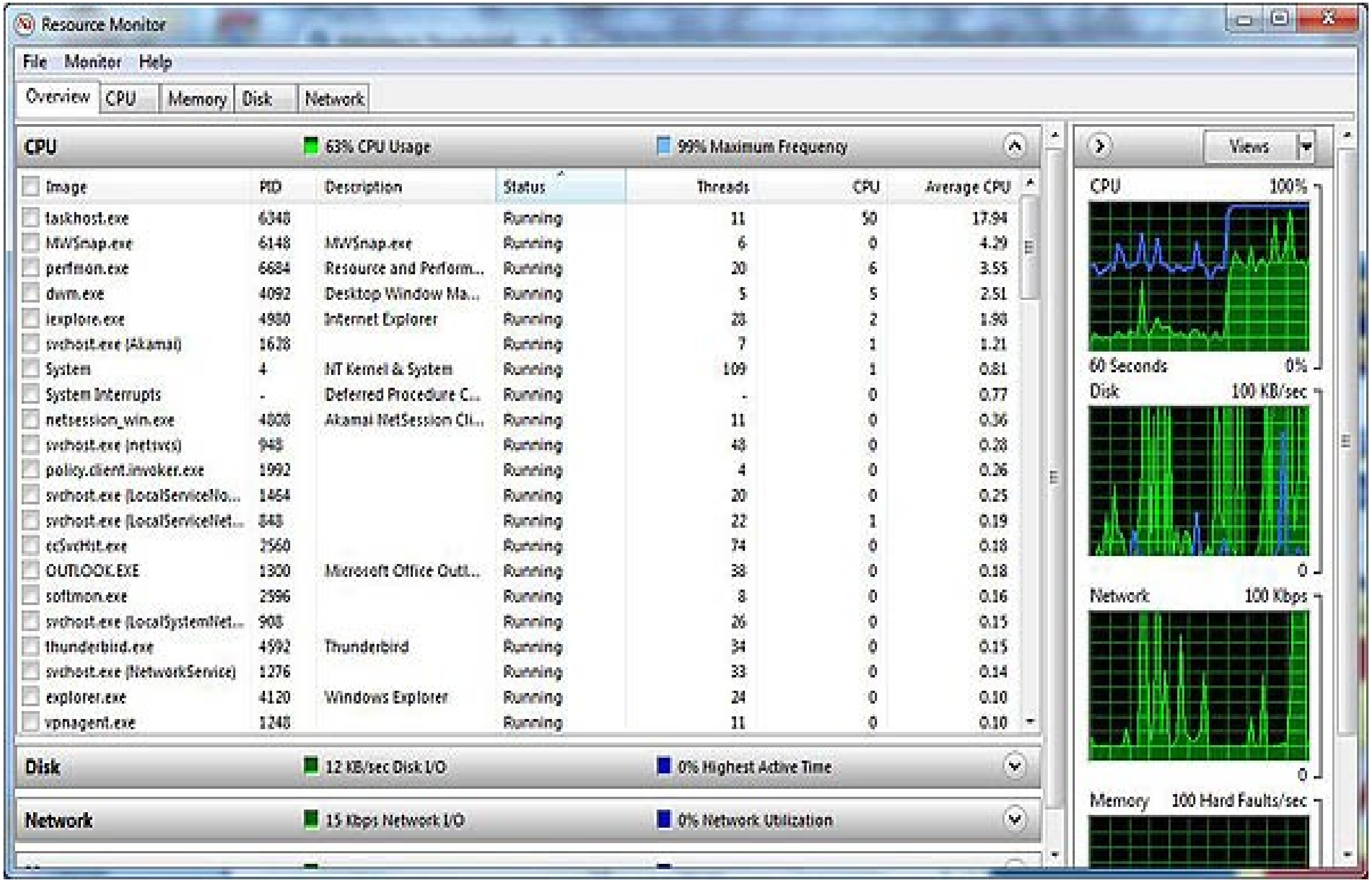
- a thread maintains its own:
  - Stack pointer
  - Registers
  - Scheduling properties (such as policy or priority)
  - Set of pending and blocked signals
  - Thread specific data.

What is a thread(3/3)



- in summary, in the UNIX environment a thread:
  - Exists within a process and uses the process resources
  - Has its own independent flow of control as long as its parent process exists and the OS supports it
  - Duplicates only the essential resources it needs to be independently schedulable
  - May share the process resources with other threads that act equally independently (and dependently)
  - Dies if the parent process dies - or something similar
  - Is "lightweight" because most of the overhead has already been accomplished through the creation of its process.

MS windows using threads



CPU 🟩 63% CPU Usage 🟦 99% Maximum Frequency

□ Image	PID	Description	Status	Threads	CPU	Average CPU
□ taskhost.exe	6348		Running	11	50	17.94
□ MWSnap.exe	6148	MWSnap.exe	Running	6	0	4.29
□ perfmon.exe	6684	Resource and Perform...	Running	20	6	3.55
□ dwm.exe	4092	Desktop Window Ma...	Running	5	5	2.51
□ iexplore.exe	4980	Internet Explorer	Running	28	2	1.98
□ svchost.exe (Akamai)	1628		Running	7	1	1.21
□ System	4	NT Kernel & System	Running	109	1	0.81
□ System Interrupts	-	Deferred Procedure C...	Running	-	0	0.77
□ netsession_win.exe	4808	Akamai NetSession Cli...	Running	11	0	0.36
□ svchost.exe (netsvcs)	948		Running	48	0	0.28
□ policy.client.invoker.exe	1992		Running	4	0	0.26
□ svchost.exe (LocalServiceNo...	1464		Running	20	0	0.25
□ svchost.exe (LocalServiceNet...	848		Running	22	1	0.19
□ ccSvcHst.exe	2560		Running	74	0	0.18
□ OUTLOOK.EXE	1300	Microsoft Office Outl...	Running	38	0	0.18
□ softmon.exe	2996		Running	8	0	0.16
□ svchost.exe (LocalSystemNet...	908		Running	26	0	0.15
□ thunderbird.exe	4592	Thunderbird	Running	34	0	0.15
□ svchost.exe (NetworkService)	1276		Running	33	0	0.14
□ explorer.exe	4120	Windows Explorer	Running	24	0	0.10
□ vpnagent.exe	1248		Running	11	0	0.10

Disk 🟩 12 KB/sec Disk I/O 🟦 0% Highest Active Time

Network 🟩 15 Kbps Network I/O 🟦 0% Network Utilization

Server Thread example.c

```
    .
    .
    .
// When accepting a new connection create a thread to handle it
While (1){
    client_sock=accept(socket_desc,(struct sockaddr*)&client(socklen_t*)&c))
    if( pthread_create( &thread , NULL , conn_handler,(void*) new_sock) <0){
            perror("could not create thread");
            return 1;
    }
    .
    .
    .
}

void *connection_handler(void *socket_desc){
      char  sendBuff[100], client_message[2000];

      while((n=recv(sock,client_message,2000,0))>0){
         send(sock,client_message,n,0);
      }
}
```

- Due to Recv been a blocking function in order to keep the server functioning and accepting new connections an independent thread is created to serve the client which just connected.
- More thread examples and various problems on the .c and python files. Look both

Server thread python example

```
while 1:
    #accept connections from outside
    (clientsocket,address)=serversocket.accept()
    #now do something with the clientsocket
    ct = thread(connection_handler)
    ct.run()

def connection_handler:
        .
        .
        .
```