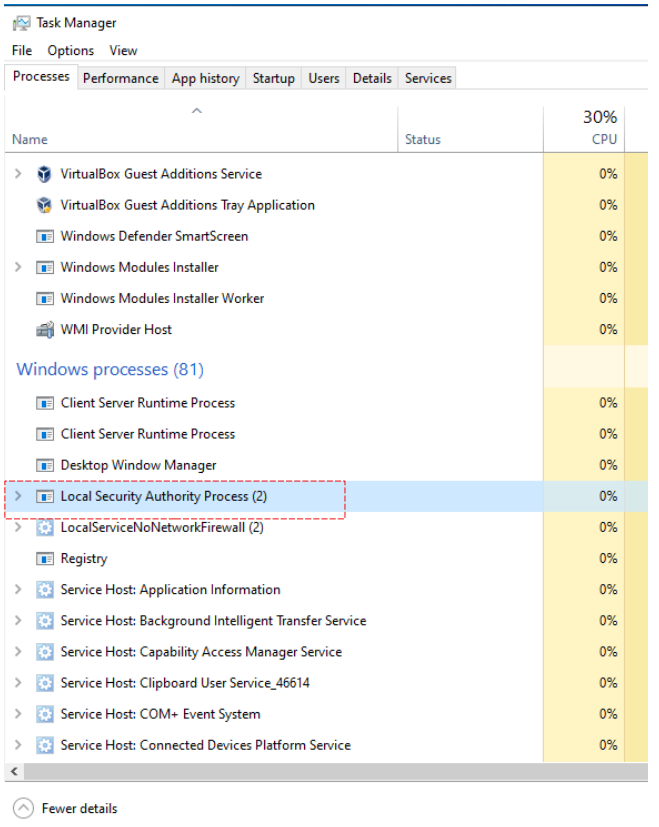
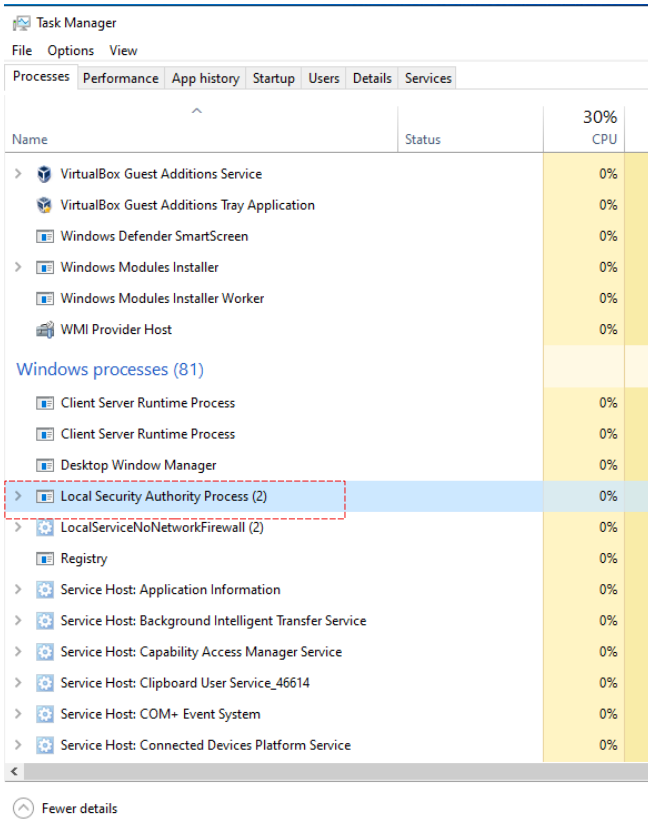
Program code: get the process id of lsass.exe

```
#include <windows.h>
#include <tlhelp32.h>
#include <stdio.h>

int main() {

    const char *targetProcName= "lsass.exe";
    int processID = 0;

  // snapshot of all processes in the system
  HANDLE processSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);

  // initializing process entry structure
  PROCESSENTRY32 processEntry;
  processEntry.dwSize = sizeof(PROCESSENTRY32);

  // info about first process encountered in a system snapshot
   BOOL operationResult  = Process32First(processSnapshot, &processEntry);

  printf("[*] Scanning for process: %s\n", targetProcName);

  // retrieve information about the processes
  while (operationResult) {

    // if we find the process: return process ID
    if (strcmp(targetProcName, processEntry.szExeFile) == 0) {
      printf("[+] Match found: %s (PID: %lu)\n", processEntry.szExeFile, processEntry.th32ProcessID);
      processID = processEntry.th32ProcessID;
      break;
    }

    operationResult = Process32Next(processSnapshot, &processEntry);
  }

  // closes an open handle (CreateToolhelp32Snapshot)
  CloseHandle(processSnapshot);

    return processID;
}
```



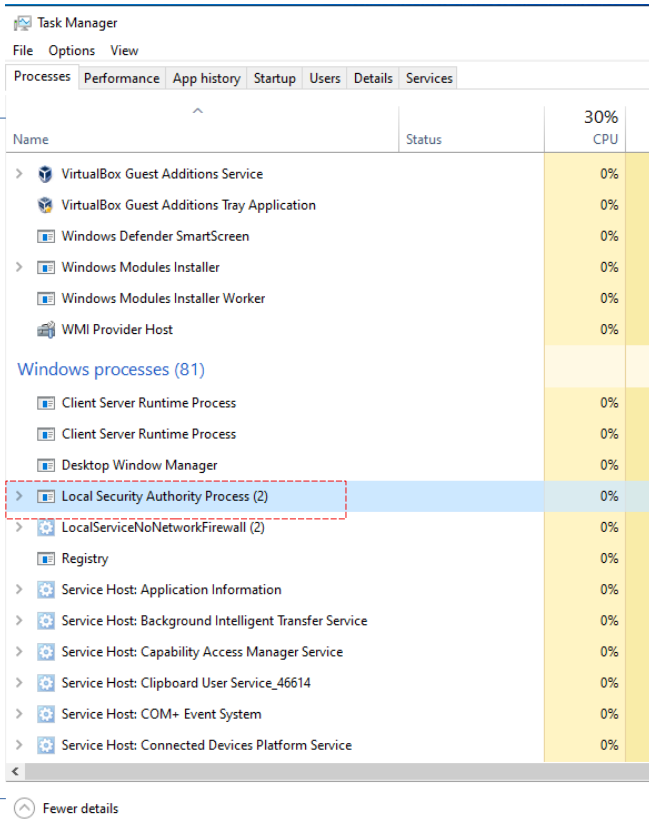
Program code: get the process id of lsass.exe

```
#include <windows.h>
#include <tlhelp32.h>
#include <stdio.h>

int main() {

    const char *targetProcName= "lsass.exe";
    int processID = 0;

  // snapshot of all processes in the system
  HANDLE processSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);

  // initializing process entry structure
  PROCESSENTRY32 processEntry;
  processEntry.dwSize = sizeof(PROCESSENTRY32);

  // info about first process encountered in a system snapshot
   BOOL operationResult  = Process32First(processSnapshot, &processEntry);

  printf("[*] Scanning for process: %s\n", targetProcName);

  // retrieve information about the processes
  while (operationResult) {

    // if we find the process: return process ID
    if (strcmp(targetProcName, processEntry.szExeFile) == 0) {
      printf("[+] Match found: %s (PID: %lu)\n", processEntry.szExeFile, processEntry.th32ProcessID);
      processID = processEntry.th32ProcessID;
      break;
    }

    operationResult = Process32Next(processSnapshot, &processEntry);
  }

  // closes an open handle (CreateToolhelp32Snapshot)
  CloseHandle(processSnapshot);

   return processID;
}
```



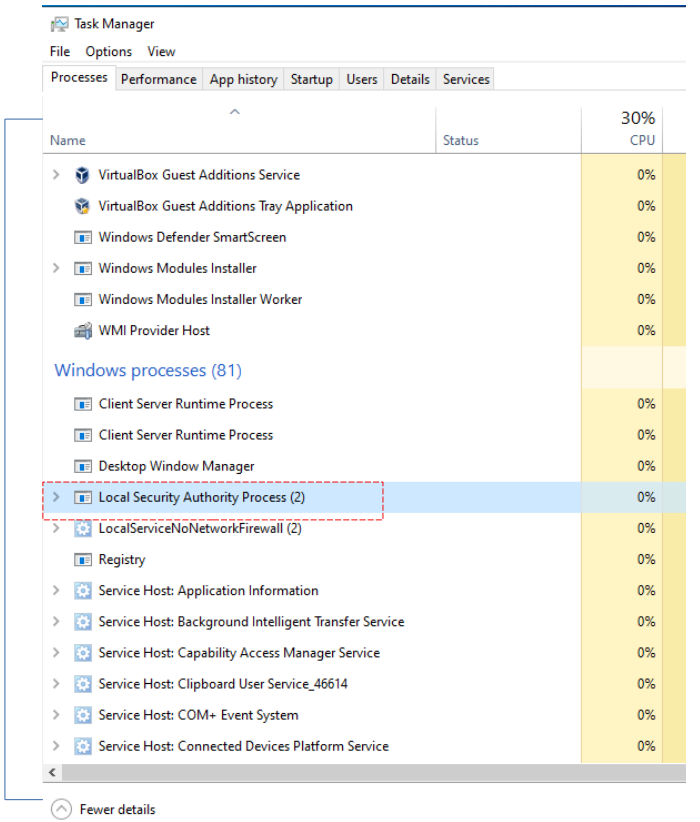
Program code: get the process id of lsass.exe

```
#include <windows.h>
#include <tlhelp32.h>
#include <stdio.h>

int main() {

    const char *targetProcName= "lsass.exe";
    int processID = 0;

  // snapshot of all processes in the system
  HANDLE processSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);

  // initializing process entry structure
  PROCESSENTRY32 processEntry;
  processEntry.dwSize = sizeof(PROCESSENTRY32);

  // info about first process encountered in a system snapshot
   BOOL operationResult  = Process32First(processSnapshot, &processEntry);

  printf("[*] Scanning for process: %s\n", targetProcName);

  // retrieve information about the processes
  while (operationResult) {

    // if we find the process: return process ID
    if (strcmp(targetProcName, processEntry.szExeFile) == 0) {
      printf("[+] Match found: %s (PID: %lu)\n", processEntry.szExeFile, processEntry.th32ProcessID);
      processID = processEntry.th32ProcessID;
      break;
    }

    operationResult = Process32Next(processSnapshot, &processEntry);
  }

  // closes an open handle (CreateToolhelp32Snapshot)
  CloseHandle(processSnapshot);

    return processID;
}
```



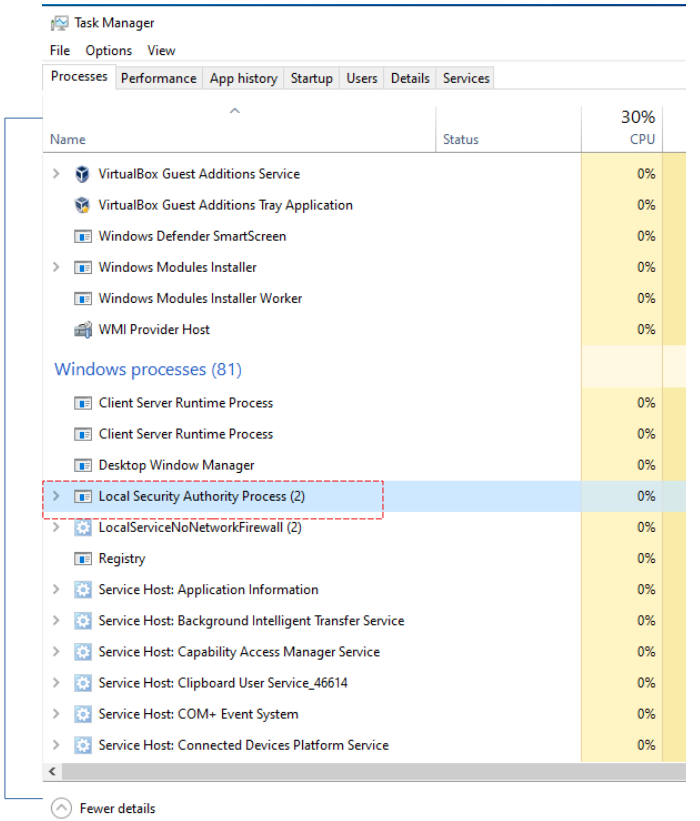
Program code: get the process id of lsass.exe

```
#include <windows.h>
#include <tlhelp32.h>
#include <stdio.h>

int main() {

    const char *targetProcName= "lsass.exe";
    int processID = 0;

  // snapshot of all processes in the system
  HANDLE processSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);

  // initializing process entry structure
  PROCESSENTRY32 processEntry;
  processEntry.dwSize = sizeof(PROCESSENTRY32);

  // info about first process encountered in a system snapshot
   BOOL operationResult  = Process32First(processSnapshot, &processEntry);

  printf("[*] Scanning for process: %s\n", targetProcName);

  // retrieve information about the processes
  while (operationResult) {

    // if we find the process: return process ID
    if (strcmp(targetProcName, processEntry.szExeFile) == 0) {
      printf("[+] Match found: %s (PID: %lu)\n", processEntry.szExeFile, processEntry.th32ProcessID);
      processID = processEntry.th32ProcessID;
      break;
    }

    operationResult = Process32Next(processSnapshot, &processEntry);
  }

  // closes an open handle (CreateToolhelp32Snapshot)
  CloseHandle(processSnapshot);

    return processID;
}
```



Program code: get the process id of lsass.exe

```
#include <windows.h>
#include <tlhelp32.h>
#include <stdio.h>

int main() {

    const char *targetProcName= "lsass.exe";
    int processID = 0;

  // snapshot of all processes in the system
  HANDLE processSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);

  // initializing process entry structure
  PROCESSENTRY32 processEntry;
  processEntry.dwSize = sizeof(PROCESSENTRY32);

  // info about first process encountered in a system snapshot
   BOOL operationResult  = Process32First(processSnapshot, &processEntry);

  printf("[*] Scanning for process: %s\n", targetProcName);

  // retrieve information about the processes
  while (operationResult) {

    // if we find the process: return process ID
    if (strcmp(targetProcName, processEntry.szExeFile) == 0) {
      printf("[+] Match found: %s (PID: %lu)\n", processEntry.szExeFile, processEntry.th32ProcessID);
      processID = processEntry.th32ProcessID;
      break;
    }

    operationResult = Process32Next(processSnapshot, &processEntry);
  }

  // closes an open handle (CreateToolhelp32Snapshot)
  CloseHandle(processSnapshot);

    return processID;
}
```



Program code: get the process id of lsass.exe

```
#include <windows.h>
#include <tlhelp32.h>
#include <stdio.h>

int main() {

    const char *targetProcName= "lsass.exe";
    int processID = 0;

  // snapshot of all processes in the system
  HANDLE processSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);

  // initializing process entry structure
  PROCESSENTRY32 processEntry;
  processEntry.dwSize = sizeof(PROCESSENTRY32);

  // info about first process encountered in a system snapshot
   BOOL operationResult  = Process32First(processSnapshot, &processEntry);

  printf("[*] Scanning for process: %s\n", targetProcName);

  // retrieve information about the processes
  while (operationResult) {

    // if we find the process: return process ID
    if (strcmp(targetProcName, processEntry.szExeFile) == 0) {
      printf("[+] Match found: %s (PID: %lu)\n", processEntry.szExeFile, processEntry.th32ProcessID);
      processID = processEntry.th32ProcessID;
      break;
    }

    operationResult = Process32Next(processSnapshot, &processEntry);
  }

  // closes an open handle (CreateToolhelp32Snapshot)
  CloseHandle(processSnapshot);

    return processID;
}
```