

جامعة المنارة

كلية:الهندسة.....

قسم: الهندسة المعلوماتية.....

اسم المقرر: نظم تشغيل 1.....

رقم الجلسة (...3...)

عنوان الجلسة

إدارة المهام وبرمجة العمليات Processes والمسالك Threads

م.فاطمة جنيدي

م.عمار مصطفى



العام الدراسي 2024/ 2025

الفصل الدراسي

الغاية من الجلسة

- ✓ التعريف بأساليب إدارة العمليات في نظام التشغيل.
- ✓ استخدام لغة البرمجة (C++) في إنشاء العمليات Processes وتغيير حالتها وإنهائها.
- ✓ كتابة شيفرات متقدمة للتعامل مع المسالك Threads.

مقدمة

العملية Process هي عبارة عن برنامج في حالة تنفيذ. تحتاج العملية إلى الموارد معينة (مثل زمن المعالج CPU والذاكرة memory وأجهزة الدخل والخرج) لكي تنجز مهمتها، ويتم تخصيص هذه الموارد للعملية طالما أنها في حالة تنفيذ.

تعتبر العملية الوحدة التشغيلية الأساسية في أغلب نظم التشغيل، إذا يكون النظام عبارة عن مجموعة من العمليات: عمليات خاصة بالنظام تنفذ الشيفرة الخاصة بالنظام، وعمليات خاصة بالمستخدم تنفذ شيفرة المستخدم حيث يمكن أن تنفذ جميع هذه العمليات بشكل متزامن.

تدعم أنظمة التشغيل الحديثة العمليات التي تملك العديد من مسالك (مسارات) التحكم threads of control. إذ يمكن تشغيل هذه المسالك على التوازي في الأنظمة التي تحتوي على مراكز معالجة متعددة.

إن أحد أهم جوانب نظام التشغيل هو كيفية جدولة المسالك على مراكز المعالجة المتاحة إذ تتوفر عدة خيارات لتصميم جدولة وحدة المعالجة المركزية.

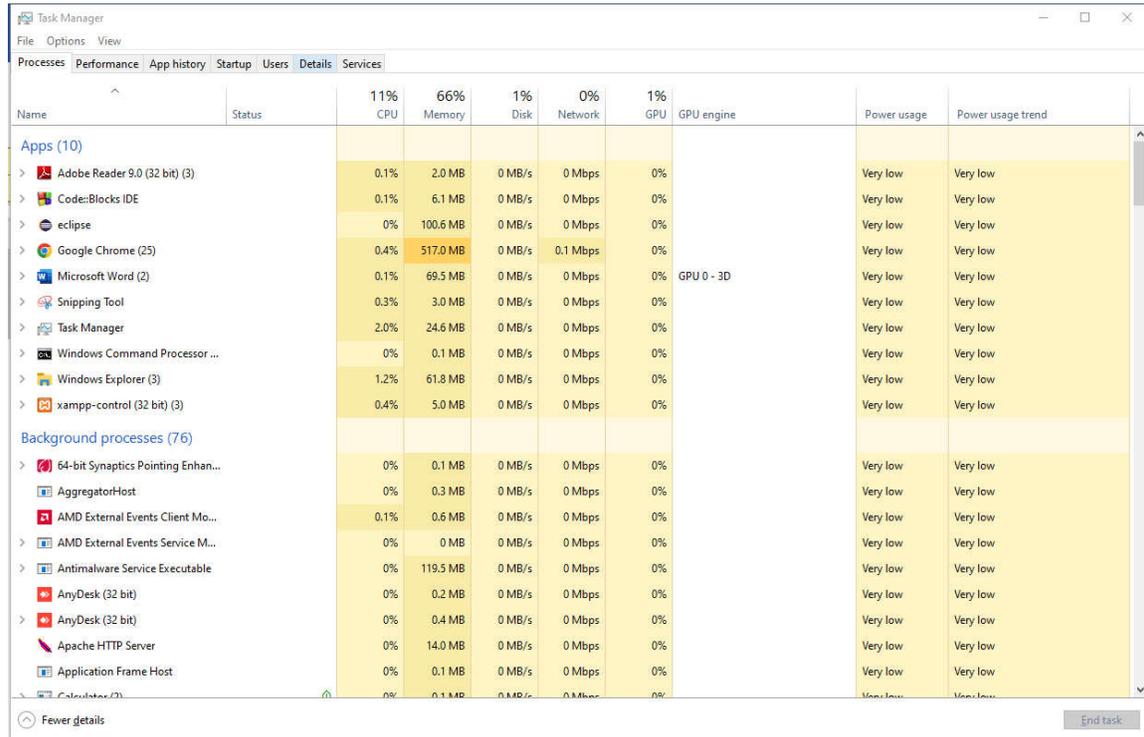
تعرفنا في القسم النظري من هذا المقرر على المفاهيم البنيوية الخاصة بالعمليات والمسارات والجدولة، ونخصص حيزاً في جلساتنا العملية (الحالية والقادمة) للتعرف على بعض الممارسات العملية على أنظمة التشغيل windows و Linux التي تتيح لنا إدارة العمليات وتنظيم عمليات الجدولة، ثم نخطو خطوة متقدمة إضافية للتعرف على كيفية كتابة برامج بلغة C (أو C++) لإنشاء عمليات ومسالك وتحقيق بعض الأمثلة حول خوارزميات الجدولة المختلفة.

إدارة العمليات في نظام التشغيل Windows:

استخدام الأدوات الرسومية: تتوفر في نظام التشغيل Windows أداة رسومية GUI خاصة بإدارة العمليات أو المهام تدعى Windows Task Manager ويمكن تشغيلها بأساليب عديدة، نذكر منها:

- استخدام أداة تشغيل البرامج بالضغط على تراكب المفاتيح (أيقونة الويندوز + R) وكتابة الأمر taskmgr.
- من محث الأوامر command prompt كتابة الأمر taskmgr.
- بالضغط على تراكب المفاتيح ctrl+alt+del ومن ثم اختيار task manager.
- من الأداة msconfig (التي تعرفنا عليها في الجلسة السابقة)، اختيار التبويب tools ثم اختيار الأداة Task manager والضغط على الزر launch.

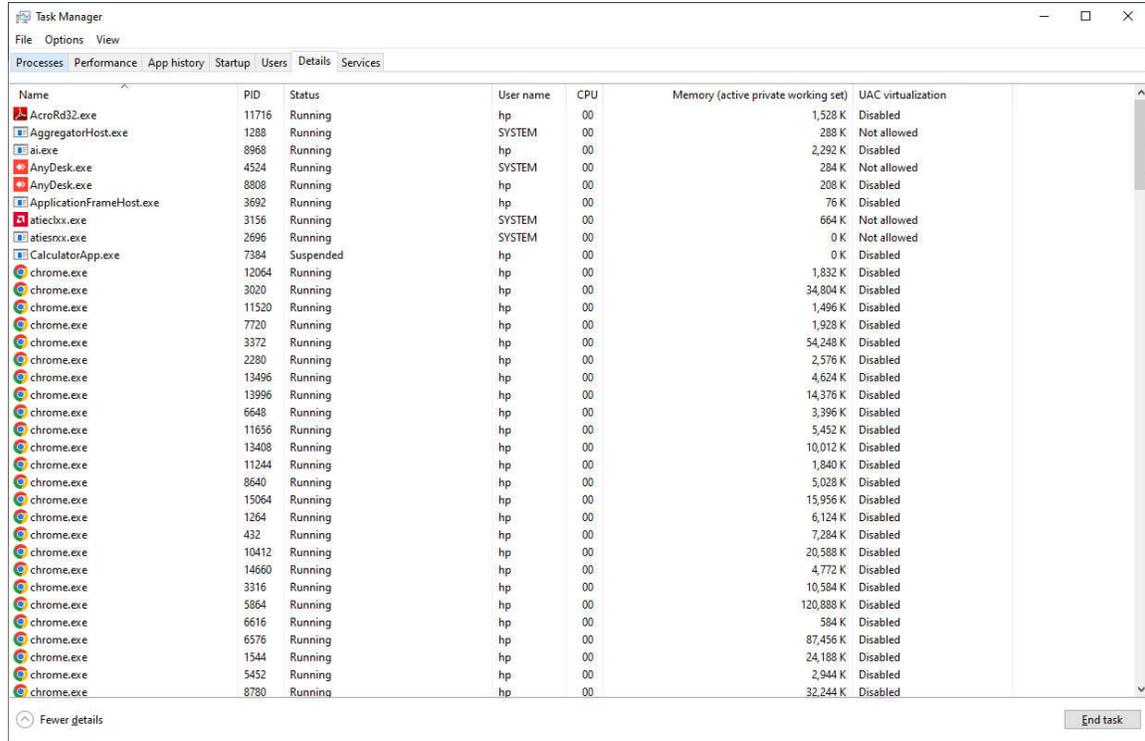
في جميع هذه الحالات تظهر لدينا النافذة التالية:



Name	Status	11% CPU	66% Memory	1% Disk	0% Network	1% GPU	GPU engine	Power usage	Power usage trend
Apps (10)									
Adobe Reader 9.0 (32 bit) (3)		0.1%	2.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Code::Blocks IDE		0.1%	6.1 MB	0 MB/s	0 Mbps	0%		Very low	Very low
eclipse		0%	100.6 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Google Chrome (25)		0.4%	517.0 MB	0 MB/s	0.1 Mbps	0%		Very low	Very low
Microsoft Word (2)		0.1%	69.5 MB	0 MB/s	0 Mbps	0%	GPU 0 - 3D	Very low	Very low
Snipping Tool		0.3%	3.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Task Manager		2.0%	24.6 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Windows Command Processor ...		0%	0.1 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Windows Explorer (3)		1.2%	61.8 MB	0 MB/s	0 Mbps	0%		Very low	Very low
xampp-control (32 bit) (3)		0.4%	5.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Background processes (76)									
64-bit Synaptics Pointing Enhanc...		0%	0.1 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Aggregator-Host		0%	0.3 MB	0 MB/s	0 Mbps	0%		Very low	Very low
AMD External Events Client Mo...		0.1%	0.6 MB	0 MB/s	0 Mbps	0%		Very low	Very low
AMD External Events Service M...		0%	0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Antimalware Service Executable		0%	119.5 MB	0 MB/s	0 Mbps	0%		Very low	Very low
AnyDesk (32 bit)		0%	0.2 MB	0 MB/s	0 Mbps	0%		Very low	Very low
AnyDesk (32 bit)		0%	0.4 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Apache HTTP Server		0%	14.0 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Application Frame Host		0%	0.1 MB	0 MB/s	0 Mbps	0%		Very low	Very low
Calculator (2)		0%	0.1 MB	0 MB/s	0 Mbps	0%		Very low	Very low

تظهر هذه النافذة قائمة بالعمليات (عمليات النظام وعمليات الخلفية) والبرامج والتطبيقات التي هي قيد التشغيل مع معلومات كاملة عن كل عملية تشمل مقدار استهلاكها من زمن المعالج، الذاكرة، الأقراص التخزينية ... إلخ.

يمكن اختيار التبويب تفاصيل details لعرض نفس القائمة ولكن مع تفاصيل مختلفة (أهمها حالة العملية ومعرف العملية PID) كما في الشاشة التالية:

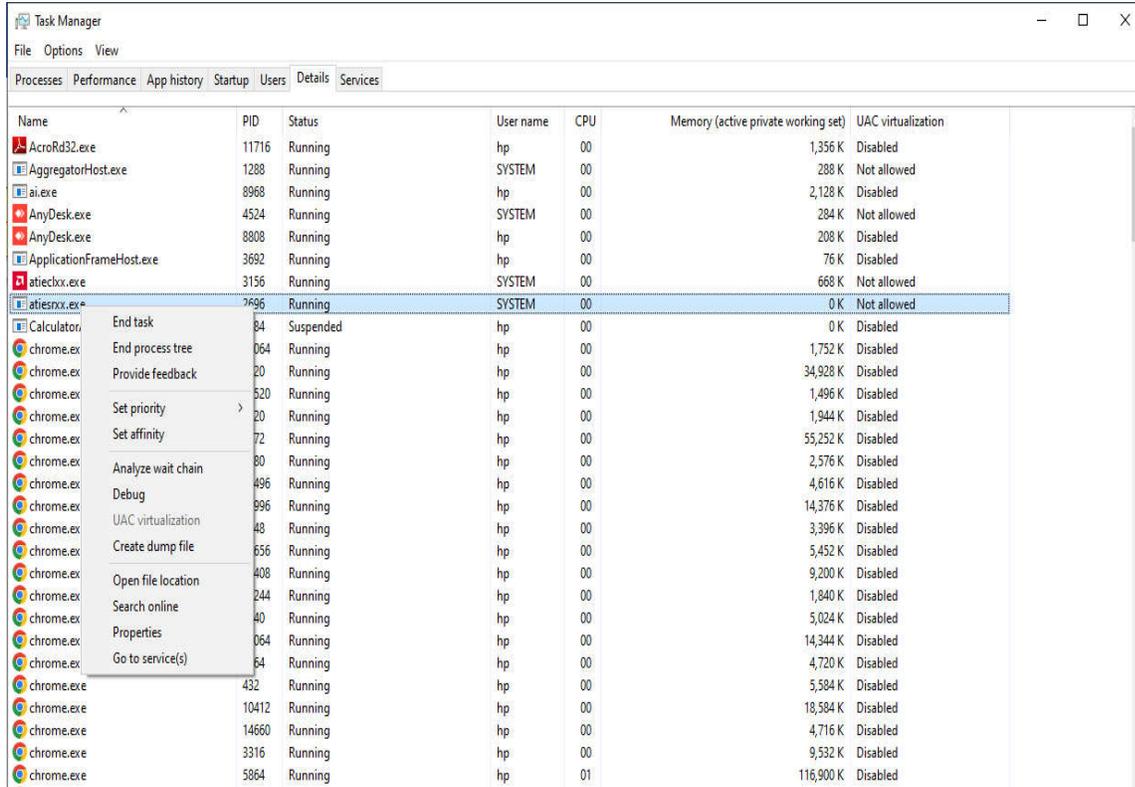


Name	PID	Status	User name	CPU	Memory (active private working set)	UAC virtualization
AcroRd32.exe	11716	Running	hp	00	1,528 K	Disabled
AggregatorHost.exe	1288	Running	SYSTEM	00	288 K	Not allowed
al.exe	8968	Running	hp	00	2,292 K	Disabled
AnyDesk.exe	4524	Running	SYSTEM	00	284 K	Not allowed
AnyDesk.exe	8808	Running	hp	00	208 K	Disabled
ApplicationFrameHost.exe	3692	Running	hp	00	76 K	Disabled
atieclxx.exe	3156	Running	SYSTEM	00	664 K	Not allowed
atiesxxx.exe	2696	Running	SYSTEM	00	0 K	Not allowed
CalculatorApp.exe	7384	Suspended	hp	00	0 K	Disabled
chrome.exe	12064	Running	hp	00	1,832 K	Disabled
chrome.exe	3020	Running	hp	00	34,804 K	Disabled
chrome.exe	11520	Running	hp	00	1,496 K	Disabled
chrome.exe	7720	Running	hp	00	1,928 K	Disabled
chrome.exe	3372	Running	hp	00	54,248 K	Disabled
chrome.exe	2280	Running	hp	00	2,576 K	Disabled
chrome.exe	13496	Running	hp	00	4,624 K	Disabled
chrome.exe	13996	Running	hp	00	14,376 K	Disabled
chrome.exe	6648	Running	hp	00	3,396 K	Disabled
chrome.exe	11656	Running	hp	00	5,452 K	Disabled
chrome.exe	13408	Running	hp	00	10,012 K	Disabled
chrome.exe	11244	Running	hp	00	1,840 K	Disabled
chrome.exe	8640	Running	hp	00	5,028 K	Disabled
chrome.exe	15064	Running	hp	00	15,956 K	Disabled
chrome.exe	1264	Running	hp	00	6,124 K	Disabled
chrome.exe	432	Running	hp	00	7,284 K	Disabled
chrome.exe	10412	Running	hp	00	20,588 K	Disabled
chrome.exe	14660	Running	hp	00	4,772 K	Disabled
chrome.exe	3316	Running	hp	00	10,584 K	Disabled
chrome.exe	5864	Running	hp	00	120,888 K	Disabled
chrome.exe	6616	Running	hp	00	584 K	Disabled
chrome.exe	6576	Running	hp	00	87,456 K	Disabled
chrome.exe	1544	Running	hp	00	24,188 K	Disabled
chrome.exe	5452	Running	hp	00	2,944 K	Disabled
chrome.exe	8780	Running	hp	00	32,244 K	Disabled

تأتي أهمية هذه الأداة من الاستخدامات المختلفة التي تتيحها ومنها:

- 1- استخدامات لأغراض التشخيص، إذ أنها تمكننا من معرفة العمليات التي تستهلك كما كبيراً من الموارد (معالج، ذاكرة، أقراص، طاقة إلخ) وبالتالي القيام بالإجراء المناسب في هذه الحالة سواء بإيقاف هذه العملية (إنهاؤها) أو تغيير حالتها أو إزالة البرنامج المسبب للمشاكل.
- 2- استكشاف البرمجيات الضارة والفيروسات (التي تعمل في الخلفية) واتخاذ الإجراء المناسب تجاهها.
- 3- استكشاف محاولات الاختراق عبر الشبكة.
- 4- يمكن أن تستخدم الأداة لإعطاء أولوية ما لعملية مستعجلة، وذلك من خلال إيقاف العمليات التي تشكل عائقاً أما تنفيذها بسرعة أو تغيير حالتها إلى وضع الإنتظار.

للتعامل مع أي عملية وتنفيذ إجراءات إدارية عليها، يمكن الضغط بالزر اليميني على هذه العملية لتظهر قائمة بالعمليات المتاحة (كما في الشكل):



Name	PID	Status	User name	CPU	Memory (active private working set)	UAC virtualization
AcroRd32.exe	11716	Running	hp	00	1,356 K	Disabled
AggregatorHost.exe	1288	Running	SYSTEM	00	288 K	Not allowed
ai.exe	8968	Running	hp	00	2,128 K	Disabled
AnyDesk.exe	4524	Running	SYSTEM	00	284 K	Not allowed
AnyDesk.exe	8808	Running	hp	00	208 K	Disabled
ApplicationFrameHost.exe	3692	Running	hp	00	76 K	Disabled
aticcxxx.exe	3156	Running	SYSTEM	00	668 K	Not allowed
aticcxxx.exe	7696	Running	SYSTEM	00	0 K	Not allowed
Calculator	84	Suspended	hp	00	0 K	Disabled
chrome.exe	064	Running	hp	00	1,752 K	Disabled
chrome.exe	20	Running	hp	00	34,928 K	Disabled
chrome.exe	520	Running	hp	00	1,496 K	Disabled
chrome.exe	20	Running	hp	00	1,944 K	Disabled
chrome.exe	72	Running	hp	00	55,252 K	Disabled
chrome.exe	80	Running	hp	00	2,576 K	Disabled
chrome.exe	496	Running	hp	00	4,616 K	Disabled
chrome.exe	996	Running	hp	00	14,376 K	Disabled
chrome.exe	48	Running	hp	00	3,396 K	Disabled
chrome.exe	656	Running	hp	00	5,452 K	Disabled
chrome.exe	408	Running	hp	00	9,200 K	Disabled
chrome.exe	244	Running	hp	00	1,840 K	Disabled
chrome.exe	40	Running	hp	00	5,024 K	Disabled
chrome.exe	064	Running	hp	00	14,344 K	Disabled
chrome.exe	64	Running	hp	00	4,720 K	Disabled
chrome.exe	432	Running	hp	00	5,584 K	Disabled
chrome.exe	10412	Running	hp	00	18,584 K	Disabled
chrome.exe	14660	Running	hp	00	4,716 K	Disabled
chrome.exe	3316	Running	hp	00	9,532 K	Disabled
chrome.exe	5864	Running	hp	01	116,900 K	Disabled

نلاحظ أنه يمكننا إنهاء العملية، إنهاء العملية وجميع العمليات المرتبطة بها (أو المسارات)، تغيير أولوية تنفيذ العملية، معرفة موقع ملفات التشغيل، معرفة الخدمات المشغلة لهذه العملية إلخ.

استخدام سطر الأوامر: يمكن الحصول على المعلومات السابقة نفسها (أو تفاصيل أكثر) وتنفيذ العديد من الأفعال على العمليات من خلال استخدام واجهة سطر الأوامر CLI في النظام Windows وذلك بواسطة العديد من الأوامر نذكر منها:

الأمر **tasklist**: لهذا الأمر العديد من البارامترات وخيارات التشغيل (يمكن التعرف عليها من خلال استخدام الأمر `help tasklist`) إلا أننا سنكتفي باستخدامه في أبسط صيغته من خلال كتابة الأمر `tasklist` على محث الأوامر لنحصل على قائمة بالعمليات كما هو مبين في الشكل:

```

C:\WINDOWS\system32\cmd.exe
C:\Users\hp>tasklist

Image Name                    PID Session Name        Session#    Mem Usage
-----
System Idle Process           0 Services             0             8 K
System                        4 Services             0           3,624 K
Registry                      100 Services             0          53,292 K
smss.exe                      424 Services             0           1,096 K
csrss.exe                     644 Services             0           5,692 K
wininit.exe                   780 Services             0           6,524 K
csrss.exe                     788 Console               1           6,496 K
services.exe                  852 Services             0           9,536 K
lsass.exe                     880 Services             0          19,844 K
winlogon.exe                  964 Console               1          11,684 K
svchost.exe                   600 Services             0          34,004 K
fontdrvhost.exe              720 Services             0           2,556 K
fontdrvhost.exe              728 Console               1           9,908 K
WUDFHost.exe                 1064 Services             0          12,532 K
svchost.exe                   1100 Services             0          16,492 K
svchost.exe                   1148 Services             0          10,276 K
svchost.exe                   1272 Services             0          11,068 K
svchost.exe                   1304 Services             0           7,464 K
svchost.exe                   1316 Services             0          10,704 K
svchost.exe                   1384 Services             0          11,124 K
dwm.exe                       1516 Console               1          89,080 K
svchost.exe                   1652 Services             0           8,204 K
svchost.exe                   1776 Services             0          12,452 K
svchost.exe                   1908 Services             0          14,716 K
svchost.exe                   1992 Services             0          13,060 K

```

ملاحظة: يمكن كتابة الأمر بالصيغة `tasklist | more` لعرض المحتوى صفحة صفحة.

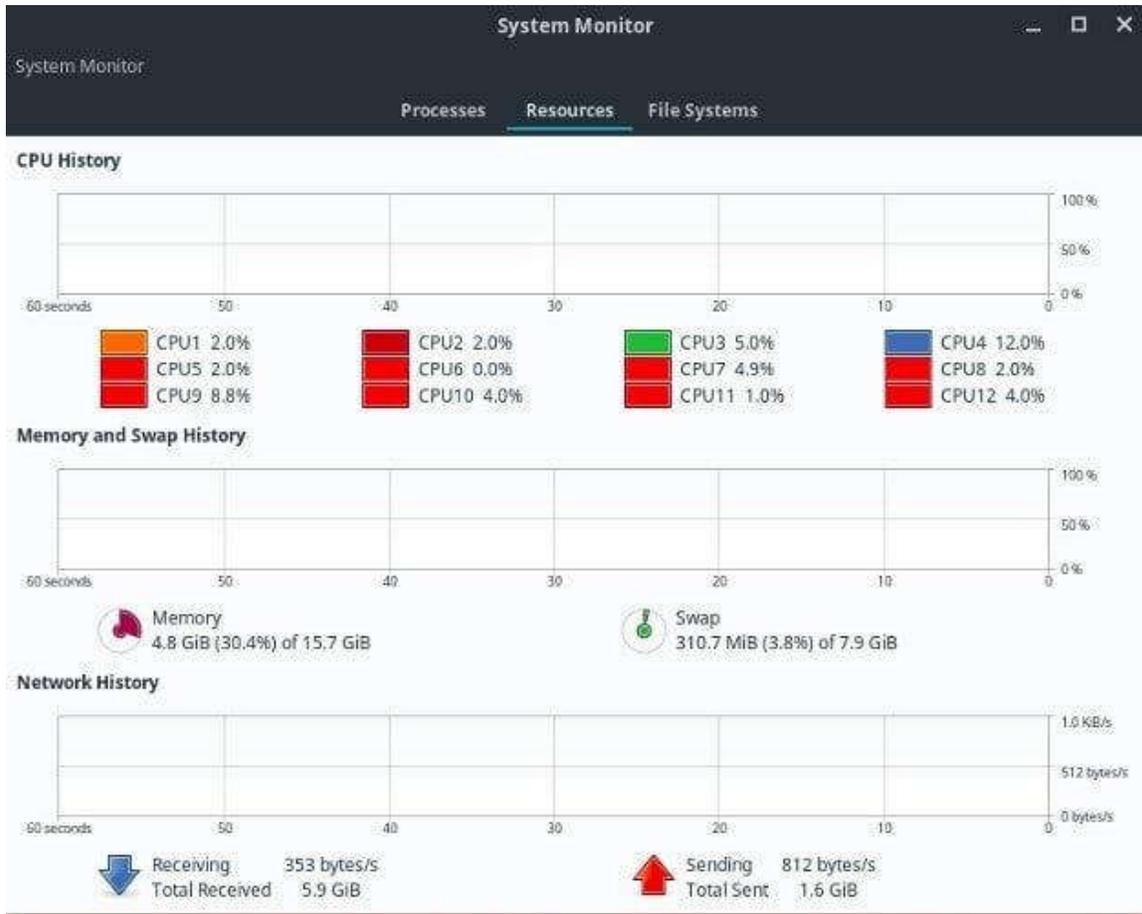
الأمر `taskkill`: يستخدم هذا الأمر من أجل إنهاء (أو إيقاف) عملية ما قيد التشغيل، لهذا الأمر العديد من الصيغ، نذكر منها:

TASKKILL /IM notepad.exe	إنهاء عملية حسب اسمها
TASKKILL /PID 1230 /PID 1241 /PID 1253 /T	إنهاء أكثر من عملية حسب رقمها مع جميع العمليات الأبناء لهذه العمليات.
TASKKILL /F /IM cmd.exe /T	إنهاء عملية حسب اسمها مع جميع أبنائها بشكل قسري

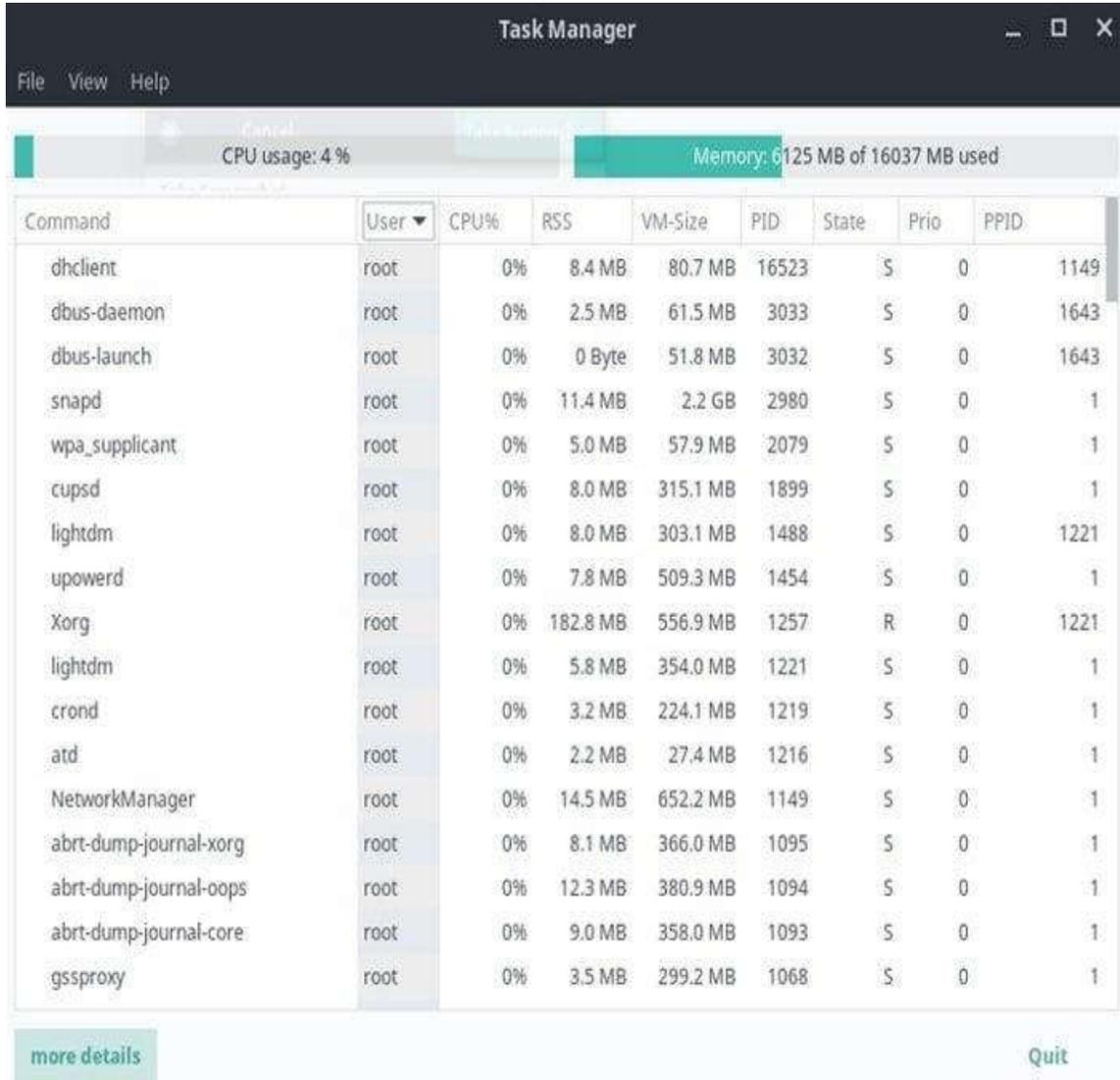
إدارة العمليات في نظام التشغيل Linux:

استخدام الأدوات الرسومية: تتوفر في نظام التشغيل Linux العديد من الأدوات الرسومية GUI الخاصة بإدارة العمليات أو المهام نذكر فيما يلي بعض هذه الأدوات:

1- الأداة **Genome System Monitor** وهي عبارة عن مدير مهام Linux خفيف الوزن وبسيط ولكنه قوي يعرض لك العديد من المعلومات المهمة حول مساحة القرص الصلب، واستخدام ذاكرة الوصول العشوائي/التبديل، وعملية التشغيل والوقت، ونشاط الشبكة، وما إلى ذلك، في شاشة عرض سهلة الفهم.



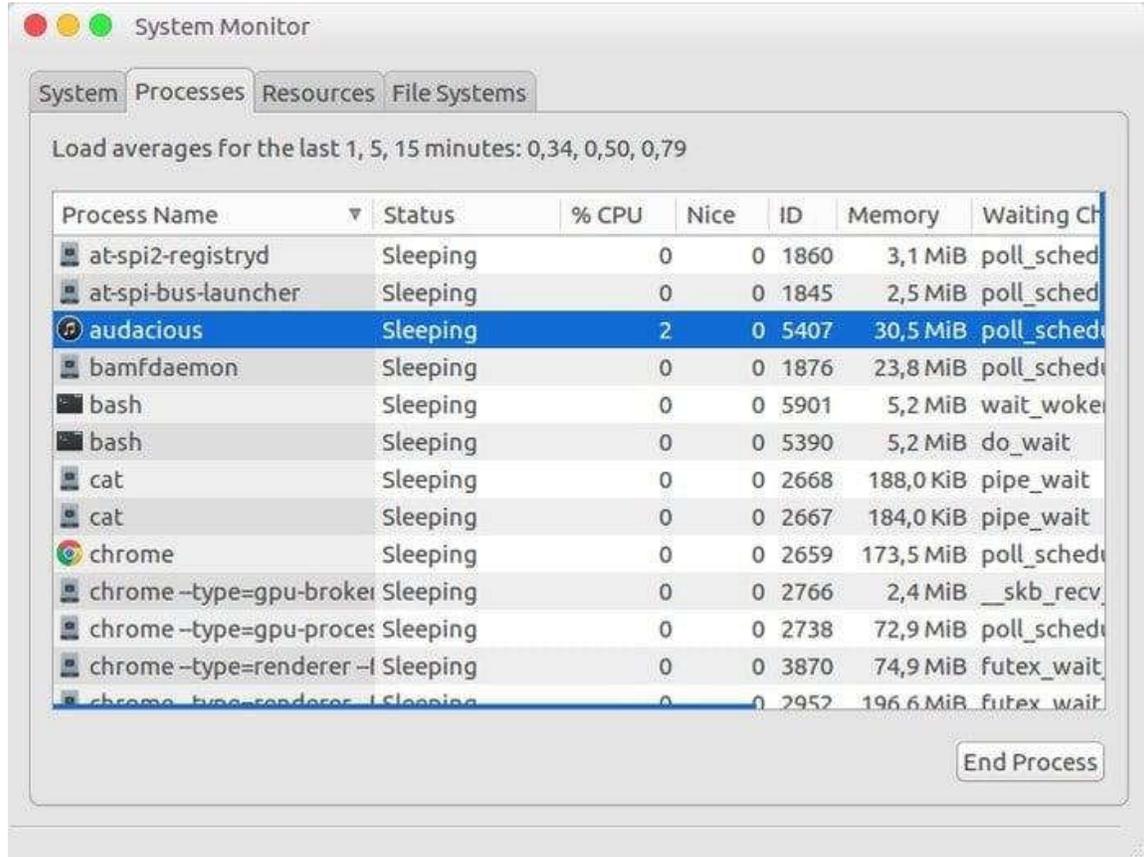
2- الأداة LXTask تعتمد هذه الأداة على مجموعة أدوات +GTK ولا تتطلب الكثير من الموارد للتشغيل على أي نظام بينما تساعدك في حل التطبيقات التي بها مشكلات وأداء بعض المهام الجذرية المحددة وعرض جدول معلومات حول أي عملية قيد التشغيل.



The screenshot shows the Task Manager window with the following data:

Command	User	CPU%	RSS	VM-Size	PID	State	Prio	PPID
dhclient	root	0%	8.4 MB	80.7 MB	16523	S	0	1149
dbus-daemon	root	0%	2.5 MB	61.5 MB	3033	S	0	1643
dbus-launch	root	0%	0 Byte	51.8 MB	3032	S	0	1643
snapd	root	0%	11.4 MB	2.2 GB	2980	S	0	1
wpa_supplicant	root	0%	5.0 MB	57.9 MB	2079	S	0	1
cupsd	root	0%	8.0 MB	315.1 MB	1899	S	0	1
lightdm	root	0%	8.0 MB	303.1 MB	1488	S	0	1221
upowerd	root	0%	7.8 MB	509.3 MB	1454	S	0	1
Xorg	root	0%	182.8 MB	556.9 MB	1257	R	0	1221
lightdm	root	0%	5.8 MB	354.0 MB	1221	S	0	1
crond	root	0%	3.2 MB	224.1 MB	1219	S	0	1
atd	root	0%	2.2 MB	27.4 MB	1216	S	0	1
NetworkManager	root	0%	14.5 MB	652.2 MB	1149	S	0	1
abrt-dump-journal-xorg	root	0%	8.1 MB	366.0 MB	1095	S	0	1
abrt-dump-journal-oops	root	0%	12.3 MB	380.9 MB	1094	S	0	1
abrt-dump-journal-core	root	0%	9.0 MB	358.0 MB	1093	S	0	1
gssproxy	root	0%	3.5 MB	299.2 MB	1068	S	0	1

3- الأداة Mate System Monitor وهي الأداة الافتراضية الموجودة في نظام Linux.



استخدام سطر الأوامر: تتوفر في نظام التشغيل Linux العديد من الأوامر الهامة نذكر منها:

1- الأمر **top**: يعد أحد أفضل مدراء مهام Linux المتاحين. يمكنك استخدام أداة إدارة المهام هذه في كل توزيع Linux باستخدام Terminal. باعتبارك مسؤول نظام Linux، باستخدام الأداة "top"، يمكنك الحصول على معلومات مهمة متنوعة حول نظامك، مثل إجمالي عدد العمليات الجارية، واستخدام وحدة المعالجة المركزية، واستخدام SWAP، وموارد ذاكرة الوصول العشوائي المجانية والمستخدم، وما إلى ذلك.

```
top - 11:36:04 up 1 day, 22:51, 2 users, load average: 0.06, 0.11, 0.09
Tasks: 141 total, 1 running, 139 sleeping, 0 stopped, 1 zombie
Cpu(s): 0.7%us, 0.5%sy, 0.0%ni, 98.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1021108k total, 982904k used, 38204k free, 134576k buffers
Swap: 2046968k total, 0k used, 2046968k free, 599576k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
25454	root	20	0	397m	107m	13m	S	2.3	10.8	25:19.18	skype
269	root	20	0	0	0	0	S	0.3	0.0	0:51.24	scsi_ah_1
1	root	20	0	2872	1400	1200	S	0.0	0.1	0:00.89	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.16	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:51.23	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:00.33	watchdog/0
7	root	RT	0	0	0	0	S	0.0	0.0	0:00.10	migration/1
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
9	root	20	0	0	0	0	S	0.0	0.0	6:44.34	ksoftirqd/1
10	root	RT	0	0	0	0	S	0.0	0.0	0:00.27	watchdog/1
11	root	20	0	0	0	0	S	0.0	0.0	0:04.05	events/0
12	root	20	0	0	0	0	S	0.0	0.0	0:04.87	events/1
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cgroup
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khelper
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	netns
16	root	20	0	0	0	0	S	0.0	0.0	0:00.00	async/mgr
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pm
18	root	20	0	0	0	0	S	0.0	0.0	0:00.23	sync_supers

2- الأمر **htop** هو عارض عمليات تفاعلي ومدير مهام قائم على النصوص لنظام Unix. توفر هذه الأداة العديد من الخيارات المفيدة لمسؤول النظام، مثل إيقاف البرامج وإعادة تشغيلها والتحكم فيها. كما يوفر أيضاً مجموعة من المعلومات حول النظام، بما في ذلك استهلاك الذاكرة ووحدة المعالجة المركزية.

```

~/projects/htop
1 [|||||] 34.3% Aug
2 [|||||] 55.0%
3 [|||||] 43.0%
4 [|||||] 42.0%
Mem [|||||] 1.16G/7.81G
Swap [|||||] 0%

Tasks: 55, 165 thr; 3 running
Load average: 0.64 0.38 0.29
Uptime: 05:19:59
Battery: 35.5% (Running on A/C)

PID USER PRI NI UIRT RES SHR S CPU% MEM% TIME Command
5177 hishan 20 0 35020 5000 4592 S 0.0 0.1 0:00.00 gdm
5176 hishan 20 0 2952 2000 1976 S 0.0 0.0 0:00.05 /bin/dbus-daemon --config-file=/System/Settings/at-spi2/ac
5175 hishan 20 0 35020 5000 4592 S 0.0 0.1 0:00.00 gdmibus
5168 root 20 0 34456 6224 5236 S 0.0 0.1 0:02.90 /usr/lib/zipower/zipowerd
5170 root 20 0 34456 6224 5236 S 0.0 0.1 0:00.00 gdmibus
5169 root 20 0 34456 6224 5236 S 0.0 0.1 0:00.00 gdmibus
5165 hishan 20 0 177M 12896 6764 S 0.0 0.2 0:47.75 /usr/bin/pulseaudio --start --log-target=syslog
5309 hishan 20 0 177M 12896 6764 S 0.0 0.2 0:00.00 alsa-source-ALC
5308 hishan 20 0 177M 12896 6764 S 0.0 0.2 0:00.00 alsa-sink-ALC36
5180 hishan 20 0 177M 12896 6764 S 0.0 0.2 0:00.01 alsa-source-ALC
5174 hishan 20 0 177M 12896 6764 S 0.0 0.2 0:45.67 alsa-sink-ALC36
5160 hishan 20 0 32288 11616 10624 S 0.7 0.1 0:00.67 xfsettingsd
5167 hishan 20 0 32288 11616 10624 S 0.0 0.1 0:00.53 gdmibus
5159 hishan 20 0 35076 17196 14320 S 0.0 0.2 0:01.17 xfce4-power-manager
5161 hishan 20 0 35076 17196 14320 S 0.0 0.2 0:00.00 gdmibus
5150 hishan 20 0 64348 31912 22820 S 0.0 0.4 0:00.60 nm-applet
5207 hishan 20 0 64348 31912 22820 S 0.0 0.4 0:00.00 gdmibus
5146 hishan 20 0 46952 22548 16712 S 0.0 0.3 0:01.52 xfdesktop
5211 hishan 20 0 46952 22548 16712 S 0.0 0.3 0:00.53 gdmibus
5144 hishan 20 0 33156 13872 12216 S 0.0 0.2 0:00.02 thunar --daemon
5153 hishan 20 0 33156 13872 12216 S 0.0 0.2 0:00.00 gdmibus
5142 hishan 20 0 39672 21724 17008 S 0.0 0.3 0:04.26 xfce4-panel
19006 hishan 20 0 10388 8600 7012 S 0.0 0.1 0:00.14 urxvt -cr green -fn x-lode-x -fb x-lode-x -fi x-lode-x -fb
19007 hishan 20 0 8788 5088 3780 S 0.0 0.1 0:00.09 zsh

F1 help F2 setup F3 search F4 filter F5 sort F6 poll F7 nice F8 nice F9 kill F10 quit

```

3- الأمر ps والأمر pstree وهي أوامر تقليدية تتيح القيام بالعديد من أعمال إدارة المهام والعمليات.

```

PID TTY STAT TIME COMMAND
1 ? Ss 0:01 /sbin/init
2 ? S 0:00 [kthreadd]
3 ? S 0:00 [ksoftirqd/0]
5 ? S< 0:00 [kworker/0:0H]
6 ? S 0:00 [kworker/u:0]
7 ? S< 0:00 [kworker/u:0H]
8 ? S 0:00 [migration/0]
9 ? S 0:00 [rcu_bh]
10 ? S 0:01 [rcu_sched]
11 ? S 0:00 [watchdog/0]
12 ? S< 0:00 [cpuset]
13 ? S< 0:00 [khelper]
14 ? S 0:00 [kdevtmpfs]
15 ? S< 0:00 [netns]
16 ? S 0:00 [bdi-default]
17 ? S< 0:00 [kintegrityd]
18 ? S< 0:00 [kblockd]
19 ? S< 0:00 [ata_sff]
20 ? S 0:00 [khubd]
21 ? S< 0:00 [md]

```

```

-+= 00000 root [swapper]
|-+= 00001 root /sbin/init --
| | |--= 00196 root /sbin/devd
| | |--= 00230 root /usr/sbin/syslogd -s
| | |--= 00344 root sendmail: accepting connections (sendmail)
| | |--= 00348 smmsp sendmail: Queue runner@00:30:00 for /var/spool/clientmqueue
| | |--= 00354 root /usr/sbin/cron -s
| | |--= 00439 _dhcp dhclient: ed0 (dhclient)
| | | += 00391 root login [pam] (login)
| | | \-+= 00400 root -csh (csh)
| | | | |
| | | | | += 00701 root /usr/local/bin/pstree
| | | | | | \-+- 00703 root sh -c ps -axww user,pid,ppid,pgid,command
| | | | | | | \--- 00704 root ps -axww user,pid,ppid,pgid,command
| | | | | | | \--- 00702 root less
| | | | |
| | | |--= 00406 root dhclient: ed0 [priv] (dhclient)
| | | |--= 00392 root /usr/libexec/getty Pc ttyv1
| | | |--= 00393 root /usr/libexec/getty Pc ttyv2
| | | |--= 00394 root /usr/libexec/getty Pc ttyv3
| | | |--= 00395 root /usr/libexec/getty Pc ttyv4
| | | |--= 00396 root /usr/libexec/getty Pc ttyv5
| | | |--= 00397 root /usr/libexec/getty Pc ttyv6
| | | |--= 00398 root /usr/libexec/getty Pc ttyv7
| | | += 00387 root sh /etc/rc autoboot
| | | \-+- 00390 root sh /etc/rc autoboot

```

استخدام لغة C (أو C++) في إنشاء العمليات والتعامل معها:

تحتوي لغة C(C++) على العديد من المكتبات والأوامر الخاصة بالتعامل مع العمليات من الناحية البرمجية نقتصر في دراستنا الحالية على بعض هذه الأوامر، نذكر منها:

Linux	Windows	الفعل
fork()	CreateProcess()	إنشاء عملية
exit()	ExitProcess()	إنهاء عملية
wait()	WaitForSingleObject()	وضع بالانتظار

نعرض فيما يلي بعض الأمثلة، للتنفيذ الأمثل لهذه التمارين يمكن كتابتها وترجمتها باستخدام إحدى بيئات تطوير لغة C(C++) وتنفيذها في نفس الوقت الذي تكون إحدى شاشات مراقبة وإدارة المهام مفتوحة ومعاينة التغيرات التي تحصل على شاشة المراقبة.

المثال الأول (نظام Windows)

ليكن لدينا الشيفرة البرمجية التالية:

```
#include <stdio.h>
#include <windows.h>
int main(VOID)
{
    STARTUPINFO si;
    PROCESS_INFORMATION pi;
    /* allocate memory */
    ZeroMemory(&si, sizeof(si));
    si.cb = sizeof(si);
    ZeroMemory(&pi, sizeof(pi));
    /* create child process */
    if (!CreateProcess(NULL, /* use command line */
        "C://WINDOWS//system32//mspaint.exe", /* command */
        NULL, /* don't inherit process handle */
        NULL, /* don't inherit thread handle */
        FALSE, /* disable handle inheritance */
        0, /* no creation flags */
        NULL, /* use parent's environment block */
        NULL, /* use parent's existing directory */
        &si,
        &pi))
    {
        fprintf(stderr, "Create Process Failed");
        return -1;
    }
}
```



```
/* parent will wait for the child to complete */
WaitForSingleObject(pi.hProcess, INFINITE);
printf("Child Complete");
/* close handles */
CloseHandle(pi.hProcess);
CloseHandle(pi.hThread);
}
```

نلاحظ عند تنفيذ البرنامج، أنه قد تم تشغيل برنامج الرسام mspaint وتم توليد عملية ظهرت في شاشة مدير المهام.

تبين التعليقات باللون الأخضر مدلول كل بارامتر من بارامترات الأمر CreateProcess كما تبين الأوامر الملونة باللون الأحمر الأوامر المستخدمة الأمرين الأساسيين الخاصين بالتعامل مع العملية، الأول هو أمر الإنشاء والأمر الثاني هو أمر الانتظار لحين إنهاء المهمة.

المثال الثاني (نظام Linux)

ليكن لدينا الشيفرة البرمجية التالية:

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
int main()
{
    pid_t pid;
    /* fork a child process */
    pid = fork();
    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
    }
}
```

```
return 1;
}
else if (pid == 0) {           /* child process */
    execlp("/bin/ls","ls",NULL);
}
else {                         /* parent process */
    /* parent will wait for the child to complete */
    wait(NULL);
    printf("Child Complete");
}
return 0;
}
```

بالمثل نلاحظ عند تنفيذ الأمر، أنه قد تم تشغيل الأمر ls وتوليد عملية ظهرت في شاشة مدير المهام.

نهاية الجلسة