# الباب الثامن

# ٨

# أدارة المشغل والمعالجات

يعتبر موضوع أدارة المشغل والمعالجات من أكثر الموضوعات التى لاقت أهتمام دارسى نظم التشغيل ، وبرز هذا الأهتمام فى عرض الموضوع من جوانب متعددة ، تارة من ناحية مفهوم المهمة ومعالجتها على وحدة التشغيل المركزية / المشغل / المعالج / وتارة من زاوية سياسات التنسيق للمشغل... إلخ مما جعل الإلمام بالموضوع يحتم الأتفاق على عدة تعريفات تباعدت ولا أقول تضاربت ، فى الأتفاق عليها غالبية المراجع التى تناولت أدارة المشغل بالبحث والدراسة، وتشمل هذه التعريفات مايلى:

## وحدة التشغيل المركزية CPU :

### المشغل PROCESSOR المعالج:

هو ذلك الجزء من الكيان الآلى الذى يهدف إلى أجراء المعالجات الحسابية والمنطقية ويتكامل فى أدائه مع وحدة التحكم وباقى وحدات المنظومة الآلية.

## الشغلة PROCESS :

هو وحدة الإيعاز الذى يتم معالجته على المشغل فى وقت محدد بذاته وتفضل النشرات والكتب الصادرة عن IBM تسميته TASK وأحياناً يسمى خطوة STEP.

وإن أجمع كثير على أن TASK مجرد عملية حسابية واحدة مثل جمع أو طرح أو ضرب.

**البرنامج PROGRAM :**

عبارة عن وحدة متكاملة من إيعازات كتبت بلغة برمجة ، وتهدف مجموعة الإيعازات فى مجملها إلى حل مشكلة بأستخدام الحاسب.

**المهمة JOB :**

عبارة عن برنامج تقدم به كاتبة إلى منظومة الحاسب وخصصت له المنظومة موارد من الكيانين الآلى والبرمجى وأدرج فى قائمة التنفيذ وبذا يتحول البرنامج إلى مهمة.

* * *

والواقع أنه طالما أتفق على المسميات فإن ذلك سوف يساعد كثيراً على فض الأشتباك بين مختلف التعريفات على مستوى هذا الكتاب على الأقل ، وندلف إلى عرض أدارة المشغل والمعالجات بعرض مثال من الحياة .

هب أن رجلاً وهبه الله قدراً من العلم والمال وأشترى لأبنه خلال جولاته على محلات سور نادى الزمالك لعبة إلكترونية ميكانيكية من التى كان يتحسر على أمثالها فى أيامه الخوالى وطفولته المبكرة فلما أفاء الله عليه من فضله لم يحرم أبنه وعاش أحلام طفولته مرة أخرى مجسدة فى ولده ، لكن اللعبة كانت مجزأة ومعها كتيب التجميع والتشغيل ، وعند عودته إلى منزله حاملاً اللعبة طلبت منه زوجته إصلاح حنفية الماء وسخان الحمام ، لكنه قرر تحت الحاح أبنه تجميع اللعبة ، فأحضر شنطة العدة [الموارد المتاحة] وبدأ فى الإعداد للتجميع بقراءة التعليمات ثم أتم تجميع القطعة (أ) وبدأ يستعد لتجميع القطعة (ب) فإذا بصراخ أبنه يأتيه من بعيد

والدم قد غطى ملابسه إذ أصطدم بالكرسى [مقاطعة] وقرر الرجل فض يـــده مـــن اللعبة والإسراع إلى علاج أبنه ، ولما أتم إسعافه عاد يستكمل تجميع اللعبة.

معنى ذلك من وجهة نظر نظم التشغيل أن:

أ ــ الرجل يقوم بدور وحدة التشغيل المركزية.

ب ــ أن المهمة هى تجميع اللعبة كما تضم قائمة الإنتظار مهمتين أخريين.

ج ــ حدوث الجرح والصراخ هو مقاطعة على تنفيذ المهمة (أ).

ولو تصورنا أن عقل الإنسان يضم برنامج قيادة هو الذى أختـــار المهمـــة القابلة للتنفيذ [تجميع اللعبة] من بين جملة مهام [إصلاح حنفيـــة المـــاء ــ إصـــلاح السخان ــ تجميع اللعبة] من قائمة إنتظار المهام وكذلك حدد خطوات وأساليب التنفيذ مثل تجميع القطعة (أ) ثم (ب)...إلخ.

معنى ذلك أن برنامج قيادة المشغل الإنسانى يضم ثلاثة روتينات:

***الأول*** : منسق المهام JOB SCHEDULAR يختار المهمة التالية.

***الثانى*** : منسق الشغلة PROCESS SCHEDULAR.

***الثالث*** : منسق على مستوى المنظومة الإنسانية.

ونستدل من ذلك على أن تنفيذ أى مهمة على المستوى الإنسانى يمـــر مـــن خلال نظام هرمى لقيادة التنفيذ ، فعلى المستوى الأعلى للقيادة يتم إختيار المهمة من قائمة إنتظار المهام بحيث يضع الموارد المتاحة فى خدمة المهمة ثم يحدد خطـــوات التنفيذ ، وهو نفس المفهوم الذى تطبقه منظومة الحاسب.

**لاحظ** : فى المثال أن المهام الثلاث تتطلب أستخدام أدوات [عدة] تكاد تكون متطابقة ، وبالطبع كل المهام التى تنفذ على الحاسب تحتاج ذات الموارد المتاحة.

**تحويل البرنامج إلى مهمة:**

أ ــ يقدم المستخدم للمنظومة (برنامجاً) يطلب تنفيذه على الحاسب.

ب ـ يتولى مسئول التشغيل مراجعة كروت JCL وباقى كروت البرنامج مع التركيز على JCL لأنها هى التى سوف تتولى إخطار نظام التشغيل بالموارد التى يحتاجها تنفيذ البرنامج.

ج ـ يقوم نظام SPOOLING بقراءة كروت البرنامج والبيانات المرتبطة وينقلها مباشرة إلى وسائط التخزين الثانوية السريعة DASD كما يخطر برنامج مدير البيانات الذى يتولى رصد البيانات المدخلة الى النظام ويحدد الحيز المتاح لها.

د ـ يتولى روتين منسق المهام مراجعة الأعمال وقائمة الإنتظار ويختار المهمة التى سيقوم النظام على تنفيذها.

هـ ـ يتولى كذلك إخطار مدير الذاكرة لتحديد الحيز المناسب فى الذاكرة وإخطار منسق المعدات لتحديد وحدات I/O وباقى الموارد المتاحة وبعدها يتم شحن المهمة فى الذاكرة وتصبح جاهزة READY ثم تتحول إلى وضع التنفيذ RUN متى أختيرت من قائمة الإنتظار.

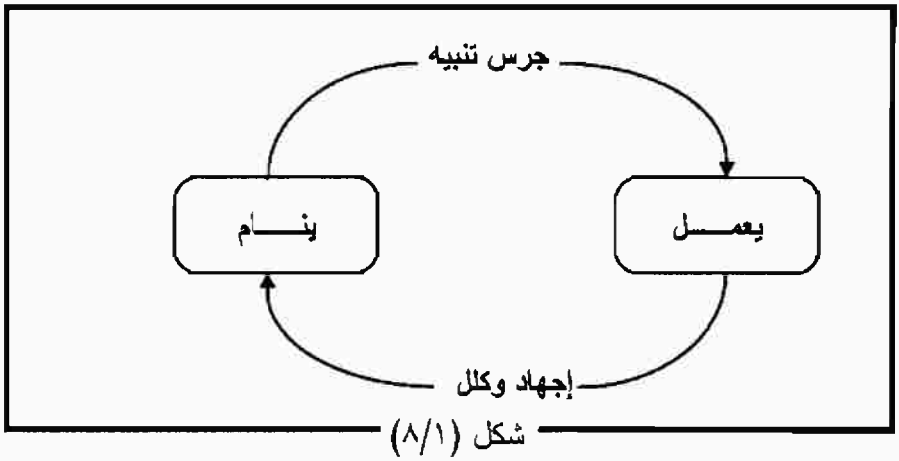
## نمذجة المعالجات:

حتى نفهم دور أدارة المشغل والمعالجات يتطلب الامر دراسة نموذج لما يتم داخل المنظومة وفى هذا الصدد نجد أنه من الأفضل البدء بدراسة نموذج مبسط بعده نضيف عدداً من العوامل المؤثرة إستكمالاً لتتبع مهام المنظومة.

### النموذج الثنائى للمعالجة:

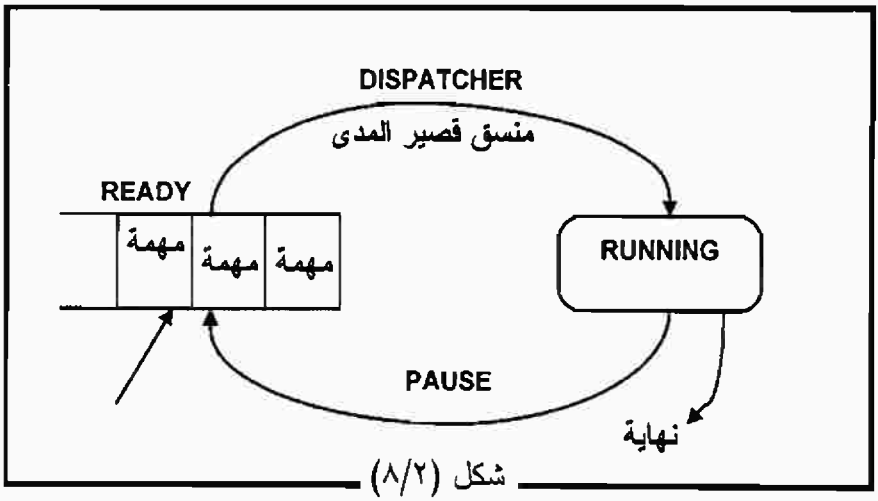
وهو نموذج يقتصر على حالتين من حالات الأداء ، كأن نقول أن الانسان فى أبسط صور أدائه يمر بحالتين هما النوم والعمل فيما يوضحه الشكل (٨/١) وأنه لكى ينتقل من حالة لأخرى يتطلب مؤثر ينقله من الحالة السابقة إلى الحالة اللاحقة ، ففى النموذج الثنائى ينهض الإنسان من النوم على صوت جرس تنبيه لينقلب حاله إلى عمل فينتج ويجتهد فإذا أصابه الكلل أرتد نائماً ، ومثل هذا النموذج غير واقعى على

الإطلاق فليس مقبولا أن يهب الإنسان من نومه ليعمل وينكب على وجهه لينام فقط ، وهذا المستوى المبسط يساعد كثيراً على فهم وإضافة وتعديل الحالات المختلفة للأداء




شكل (٨/١)

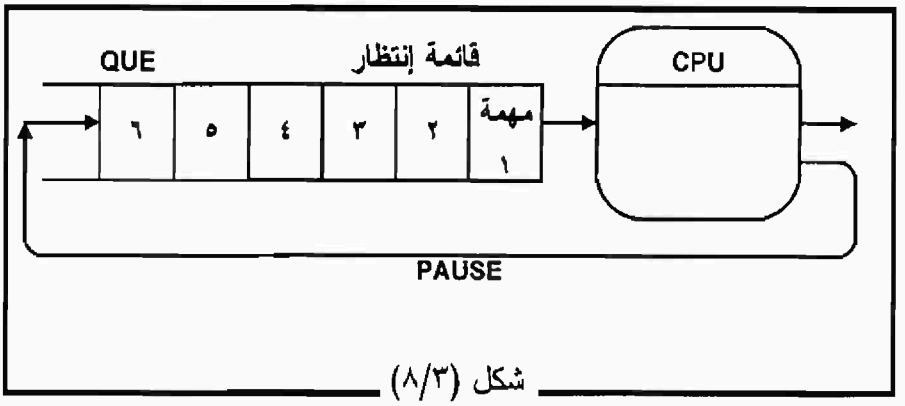
ونعود إلى المهمة التى سوف نقدمها للمنظومة ونطبق عليها النموذج الثنائى فإذا بها كالموضح بالشكل (٨/٢):




شكل (٨/٢)

ويتضح من الشكل أن المهمة الجديدة متى قبلت فى المنظومة فإنها توضع فى قائمة إنتظار ولا تدخل التنفيذ مباشرة وأنها أصبحت معرفة لنظام التشغيل ومخصصاً لها

موارد كما أوضحنا من قبل وأنها تنتقل فيما بين حالتى الإنتظار والتنفيذ بفضل منسق المهام قصير المدى DISPATCHER وخلال هذا الوقت فإن هناك مهام تخرج من المنظومة ومهام جديدة تقبل فيما يوضحه الشكل (٨/٣).




شكل (٨/٣)

ومن أبرز الأمور التى يوليها نظام التشغيل أهتماماً التعرف على المهام الجديدة خاصة موقعها فى الذاكرة ؟ لماذا .

كما يرتكز نظام التشغيل على رصد باقى الموصفات اللازمة لأدارة المهمة ووضعها فى قائمة [جدول] مكون من مجموعة من السجلات المرتبطة يسمى (قائمة توصيف المهمة) PROCESS CONTROL BLOCK وإختصاراً PCB وهى قائمة غير منظورة لمستخدم المنظومة ولا يتعامل معها وأن كانت تشبه إلى حد كبير جواز مرور مواطن أجنبى فى دولة ، وبهذه البيانات المدرجة فى PCB يقوم المنسق قصير المدى على إختيار ونقل المهمة من حالة الإنتظار إلى حالة التنفيذ / المعالجة / RUNNING / EXECUTION .

وإذا كان نظام التشغيل يرصد خصائص المهام الجديدة ويصدر لها PCB فإنه يفض الإرتباط بين المنظومة والمهام التى تم علاجها ويسقط PCB الخاصة بها من قائمة موصفات المهام LINKED LIST OF PCB ، ولعل أبرز الأمور التى تؤدى وتفضى إلى إنهاء مهمة وإستعادة الموارد المخصصة لها مايلى:

أ - نجاح تشغيل المهمة وفق متطلباتها.

ب - عدم وجود حيز مناسب فى الذاكرة.

ج - تجاوز المهمة للمدى الزمنى المحدد.

د - محاولة المهمة إجراء عمليات حسابية غير منطقية.

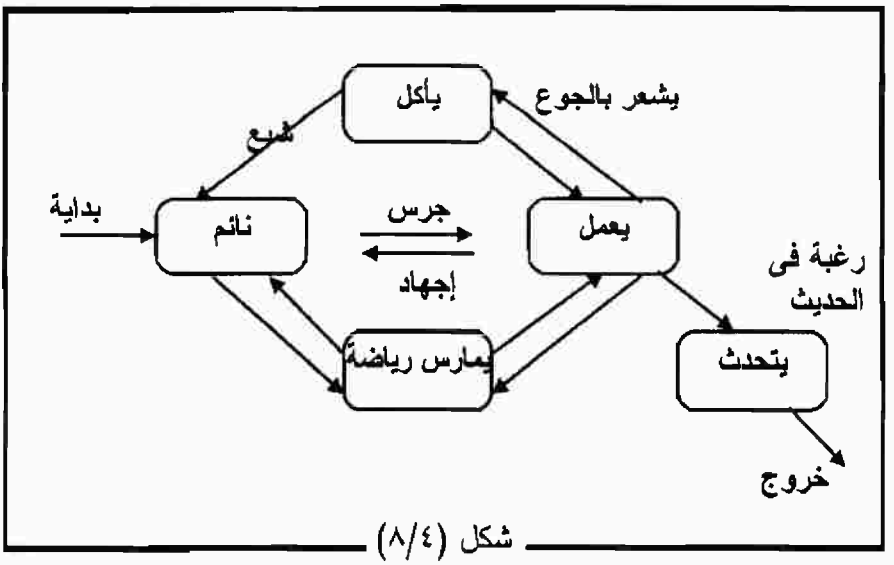
هـ - عدم قدرة المهمة على إتمام عمليات I/O.

و - محاولة المهمة اختراق مفاتيح التأمين لمهام أخرى.

ز - حدوث توقف لقلة أو شحة الموارد.

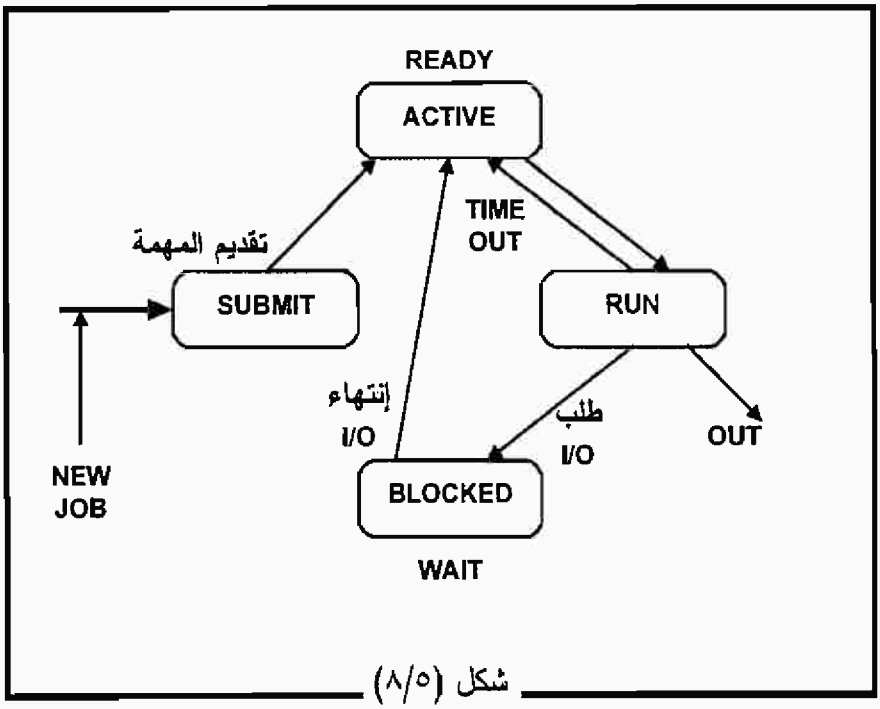
## النموذج الخماسى للمهام:

رغم بساطة النموذج الثنائى إلا أنه لا يعبر عن حالة المهمة ، شأنها فى ذلك شأن النموذج الثنائى للإنسان الذى لا يعقل أنتقاله بين حالتى النوم والعمل والعكس دون مناشط اخرى فإن شئنا أن نقترب من نموذج إنسانى أقرب للواقع لرأيناه فى الشكل (٨/٤) ، أما فى النموذج الثنائى للمهمة فنلاحظ أن المهمة لا يجرى تنفيذها بسبب عدم توافر وقت على وحدة المشغل CPU فقط وهذا غير حقيقى على الإطلاق



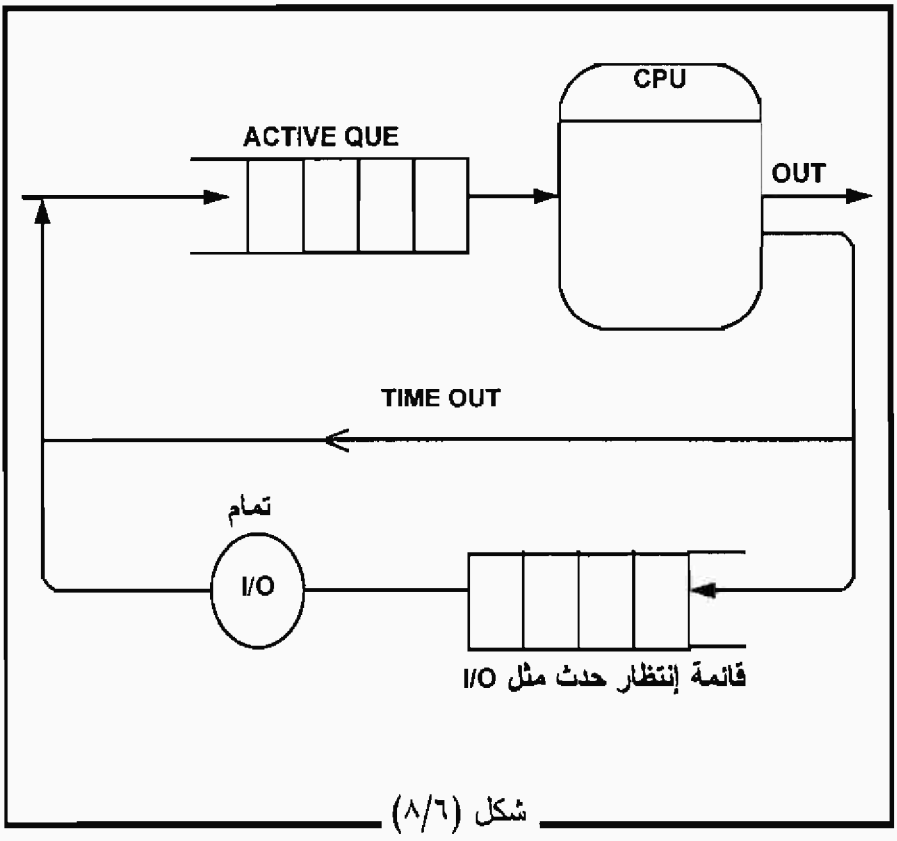

شكل (٨/٤)

فهناك أسباب عدة تعوق تنفيذ أى مهمة مثل إنتظار I/O مما يجعل أستخدام قائمة إنتظار واحدة غير ذات جدوى للمنسق قصير المدى فليس واضحاً حيالة السبب الحقيقى لدخول المهمة حالة التوقف التام أهو طلب I/O أم سبباً آخر وأيضا ليس واضحاً له المهمة التى سبقت للقائمة وبالتالى وجد من الأفضل تقسيم حالة NOT RUNING إلى قسمين حالة نشطة ACTIVE READY أى لدى المهمة جميع الموارد فيما عدا CPU والحالة الثانية من الإنتظار BLOCKED وتسمى(WAIT) تنتظر تنفيذI/O حتى تنقل من الإنتظار إلى الحالة النشطة فيما يوضحه الشكل (٨/٥) الموضح للنموذج الخماسى بعد الإضافات عليه ، ونلحظ من الشكل أن قبول المهمة من مهام الحالة النشطة يعنى ضمنا أن المنسق طويل المدى قام بدور ربة المنزل من حيث إتاحة كل الموارد المطلوبة للمهمة لتكون نشطة ومدرجة فى قائمة المهام ،



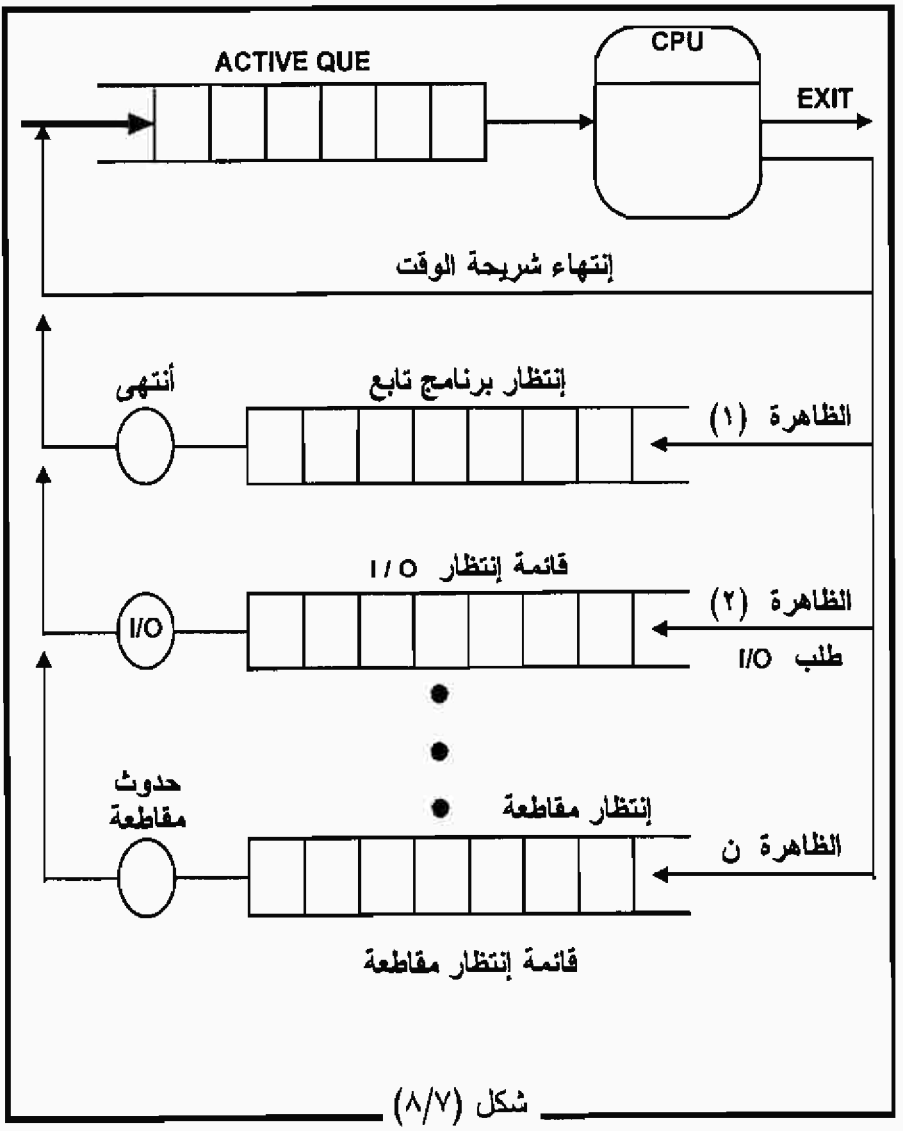

شكل (٨/٥)

ونشاط المهمة ليس دليلا على إتاحة كل الموارد بل ينقصها إتاحة وحدة التشغيل المركزية ومتى سمح لها بأستخدامها تحولت إلى حالة التنفيذ EXCEULTE وإذا طلبت المهمة I/O أنتقلت إلى حالة إنتظار WAIT أو BLOCKED وقد يكون الانتظار ليس مرده I/O بل ظاهرة أخرى EVENT خارج أهتمام وحدة التشغيل المركزية مما يتطلب أستخدام قائمتى إنتظار وليس قائمة واحدة كما فى الشكل (٨/٦) منها قائمة خاصة بالمهام التى أنتهى وقتها على المشغل وبذلك تنتقل إلى قائمة الإنتظار للمهام النشطة ACTIVE ، أما المهام التى تطلب I/O فإنها تغادر الحالة التنفيذية وتدرج فى قائمة إنتظار I/O ، والواقع هناك إحتمالات أكثر من طلب I/O




شكل (٨/٦)

ليكن عددها (ن) مما يستدعى أستخدام نفس العدد (ن) من قوائم الإنتظـــار بصـــرف النظر عن أسبقيات المهام فيما يوضحه الشكل (٨/٧) مع صياغة الأســـبقيات وفـــق قواعدها فى قوائم إنتظار خاصة.




شكل (٨/٧)

## الجداول المساعدة فى أدارة المهام:

فى إطار تعددية البرمجة يستخدم البرنامج المشرف مجموعة جداول فى السيطرة وقيادة النظام وتنظيم المهام بين مختلف الموارد الآلية والبرمجية ، وهذه البيانات تخزن على هيئة جداول TABLES وتقسم هذه الجداول إلى قسمين ، جداول ترصد بها الأعمال الجارية فى القطاعات المختلفة ، والقسم الثانى ترصد فيه المهام المؤجلة والتى قبلها النظام فيما توضحه الجداول فى الصفحة التالية ونلحظ منها مايلى:

أ ــ تحديد شامل لجميع المهام الموجودة فى قائمة إنتظار كل قطاع.

ب ــ توضيح موصفات كل مهمة جارى تنفيذها (الجدول ٢).

ج ــ تحديد المعدات الآلية لكل مهمة.

هذا بالإضافة إلى معلومات أخرى تساعد فى تتبع أداء النظام كله ومعرفة موقف كل مهمة وكل قطاع.

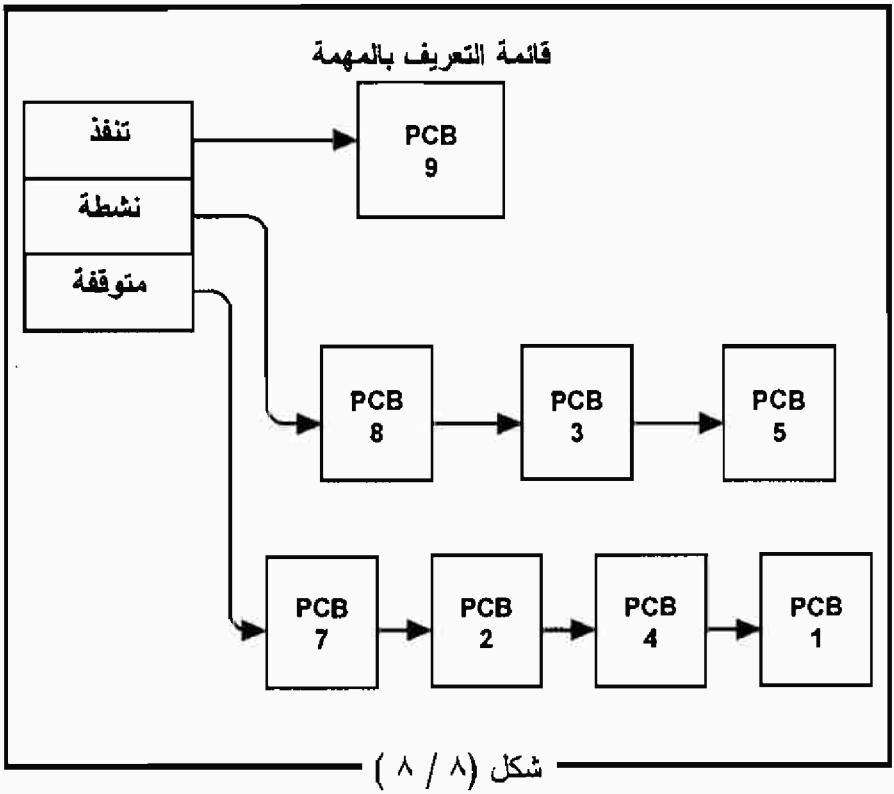
| INPUT QUE | | | |
|---|---|---|---|
| JOB NO<br>JOB NAME<br>PRIORITY<br>DEVICE | PART / 1<br>205<br>ASHRF<br>10<br>12<br>2TAP | PART / 2<br>207<br>EIMAN<br>8<br>70<br>2TAP | PART / 3<br>208<br>ZAKI<br>10<br>5<br>CARD RDR |
| PARITION NO<br>1<br>2 | QUE NO<br>205<br>207 | | |
| DEVICE TYPE<br>CARD RDR<br>PUNCH<br>PRINTER | INPUT<br>NO | OUT PUT QUE | |

| JOB IN PROGRESS | | | |
|---|---|---|---|
| PARTITION NO | 1 | 2 | 3 |
| JOB NO | 200 | 206 | 204 |
| JOB NAME | AZIZA | ALI | KIRAH |
| PERIORITY | 6 | 8 | 15 |
| MEMORY SIZE | 60 | 25 | 18 |
| DEVICE ASSIGN | TAPE 1 | ... | ... |
| STATLIS | EXECUTE | WAIT | WAIT |

| DEVICE ASSIGNED TABLE | | |
|---|---|---|
| NAME | OT WHOM | STATUS |
| CARD RDR | MAGI | EX |
| PRINTER | ALI | EX |
| TAPE 1 | ... | ... |
| TAPE 2 | ... | ... |
| CONSOL 1 | FREE | ... |
| CONSOL 2 | FREE | ... |

**بيانات التعريف بالمهمة PROCESS CONTROL BLOCK (PCB) :**

متى قبل النظام مهمة فأن منسق المهام يتولى إنشاء مـــا يســمى ببيانـــات التعريف بالمهمة PCB ، حيث تعتبر وسيلة المتابعة الوحيدة للمهمة من وقت قبولـــها الى أن يتم تنفيذها ، وتحتوى PCB على كل البيانات اللازمة لنظام التشغيـــل لأدارة المهمة.. وهى التى تشكل مجموعة الانتظار فى مختلف مراحل التنفيذ فيما يوضحـــه الشكل (٨/٨).

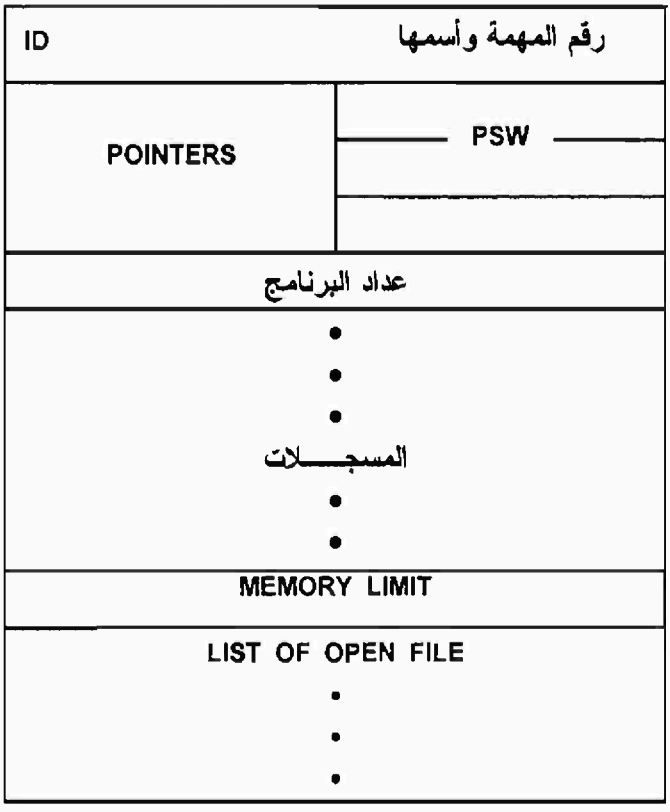



شكل (٨ / ٨)

وإذا كانت بيانات التعريف بالمهمة PCB شأنها للمهمة شأن جواز السفر لسائح والذى بأستخدام بياناته تتابعه سلطات الدولة حتى يغادر البلاد إلا أنها تعتبر أهم هيكل بيانات لنظام التشغيل ويتم تحديثها بأستمرار خلال جميع مراحل المعالجة حيث تشمل بيانات تخصيص الموارد بما فيها الذاكرة وحالة المقاطعات ورصد أداء المنظومة كلها فيما يمكن أعتبارها أنها توصف بدقة حالة نظام التشغيل ذاته ، لذا يلجأ إليها عدداً من برمجيات نظام التشغيل أثناء تواجد المهمة على المنظومة تستقى منها حالات التشغيل ورصد أدائها وتحديث بيانات المسجلات.

وتخزن PCB على مسجلات سريعة أسرع كثيراً من سرعة عناصر الذاكرة الأساسية حتى يتمكن نظام التشغيل من متابعة حالات التنفيذ.

ويوضح الشكل (٨/٩) المفردات الأساسية لقائمة البيانات PCB.




شكل (٨/٩)

محتويات قائمة بيانات مهمة

**أسم المهمة:**

ويعنى به أسم صاحب المهمة ويتم الاشارة اليه فى جداول النظام متى قبلت بالنظام.

**حالة المهمة:**

ويوضح حالة البرنامج (RUN - READY - HOLD - ...) والموارد التى سببت حالته WAIT .

**PSW :**

راجع مضمون PSW فى الباب الرابع.

**الذاكرة:**

ويقصد بها عنوان هذه المهمة فى الذاكرة الأساسية وفى حالة أسلوب الذاكرة التخيلية فما هى عناوين الصفحات والمناظر لها من أطر الصفحات.

**المسجلات:**

وتسجيل محتويات المسجلات فى حالات المقاطعة.

**الآلات:**

بيان بكل الموارد الآلية المستخدمة فى تنفيذ المهمة.

**الأسبقية:**

وتعطى وفق سياسة التشغيل FIFO أو دورة روبين... إلخ.

**بيانات محاسبية:**

يحتوى هذا الحقل على جميع البيانات اللازمة لقياس الأداء وكذلك حساب تكلفة التشغيل مثل الزمن المستهلك على وحدة التشغيل المركزية ، زمن بقاء البرنامج فى الذاكرة ، عدد صفحات البرنامج وزمن بقائها (صفحة / ثانية) على وسائط التخزين الثانوية ، وزمن إنتظار البرنامج ، وما هى الموارد البرمجية التى أستخدمت فى تنفيذ المهمة [ترجمة ــ ربط ..] عدد وحدات المدخلات / المخرجات والفترة الزمنية التى أشتغلتها والوقت المستهلك فى الإنتظار.

## تنسيق أداء المشغل:

### أهداف التنسيق:

إذا كان غاية التنسيق الأرتقاء بمعولية المنظومة وإنتاجيتها فهناك مبادئ يجب إلا تغيب عن بال المسئول المباشر عن التخطيط والتنفيذ ، وهذه المبادئ الأساسية هى:

أ ـ تحقيق أعلى إنتاجية للنظام أما بتشغيل أكبر عدد من البرامج فى الزمن المحدد لتشغيل النظام ، أو بتشغيل عدد أكبر من البرامج الصغيرة الحيز أو تنفيذ برامج خالية من المقاطعات قدر الإمكان.

ب ـ تقليل زمن الإستجابة للحد الأدنى بتنفيذ برامج الوقت الحقيقى أولا وتأجيل برامج الدفعة BATCH.

ج ـ تقليل الوقت الضائع على النظام وذلك بتنفيذ برامج الدفعة أولا ثم تنفيذ برامج الوقت الحقيقى.

د ـ تقليل وقت الإنتظار بدفع المهام إلى التنفيذ دون إنتظار متى وصلت حالة READY وإزاحتها فوراً من النظام متى تم تنفيذها.

هـ ـ زيادة كفاءة وحدة التشغيل المركزية بحيث تعمل بكامل جهدها طول الوقت وذلك بأستقبال البرامج العلمية والحسابية.

و ـ تقليص التغاير فى زمن الاستجابة بدلا من تقليص متوسط زمن الإستجابة.

ويعيب هذه الأهداف تعارضها بشدة مع بعضها البعض مما يحتاج إلى تطبيق سياسات مختلفة مع البرامج المختلفة ، ونلحظ الألتزام بهذه السياسات فى أبرز أشكالها فى مراكز الحاسبات فى الدول الغربية والتى تدار بمفهوم أقتصادى لأنها مجرد شركات لنظم المعلومات والحاسبات تقدم خدمات بالأجر وهدفها الأساسى تعظيم أرباحها وتعظيم عدد مستخدمى النظام مع جذب مستخدمين جدد بأستمرار.

## سياسات التنسيق:

يمكن النظر إلى سياسات التنسيق عبر عدة منظورات:

**أولا :** منظور طبيعة المهمة وهل هى من النوع الذى ينجز دفعة واحدة أم يتم إعادة إدخالها إلى المنظومة مرة أخرى بل مرات متعددة كما فى أنظمة

المشاركة على الوقت ونلحظ أن هذا المنظور يحتم أستخدام سياستين دون سواهما هى:

(١) من يأتى أولا يخدم أولا FCFS وهو أسلوب يناسب المعالجة بنظام الدفعة BATCH أو المستخدم الواحد لحيز متصل أو سلسلة مهام JOB STREAM.

(٢) أسلوب ROUND ROBIN (R. R) ويصلح لعدد من المهام المشاركة على الوقت التى تتطلب إعادة إدخال المهام إلى المشغل PREEMPTED.

**ثانيا :** منظور يركز على الربحية وتحقيق أعلى مكاسب لمنظومة الحاسب ، ونلحظ عدة سياسات تركز همها على هذا الهدف:

(١) الخدمة وفق أسبقية العميل.

(٢) تفضيل الخدمة الأقصر زمناً SHORTES JOB FIRST.

(٣) الخدمة وفق رغبات العميل.

(٤) الخدمة الأعلى سعراً.

(٥) تقديم الخدمة وفق خبرة المدير التنفيذى (مدير التشغيل) للمنظومة.

والواقع أن سياسات التنسيق لأعلى ربحية لا يؤخذ بها فى منظومات المعلومات المرتكنة على الحاسبات فى القطاعات الحكومية فى دول العالم الثالث وهذا أمر بالغ الخطورة لأن لأى منظومة هدفاً يجب أن تحققه فى إطار تعظيم الربحية سيان كانت الربحية منظورة أو غير منظورة فدون معيار أقتصادى لا إنضباط أو جدية لأى عمل مهما كان ونخلص من هذا العرض أن أبرز سياستى تنسيق هما:

١ ــ من يأتى أولاً يخدم أولاً 1 - FCFS

٢ ــ دوارة روبين 2 - R . R

علماً بأن FCFS هى الأبسط والأدق والأكثر عدلاً بين كل السياسات السابق الإشارة إليها إذ تعطى معياراً لجميع المهام التى يتطلب معالجتها طرحها فى النظام دفعة واحدة ، ولا تحتاج برامج التنسيق إلا لمعلومات محدودة جداً عن المهام المختلفة ، ورغم العدالة والبساطة إلا إنها قد لاترضى مستخدمى المنظومة خصوصاً المهام التى تحتاج زمناً ضئيلاً محدوداً على المشغل.

## معايير مقارنة السياسات:

هناك عدة معايير لمقارنة سياسات التشغيل منها:

### ١ ـ إتاحة وحدة التشغيل المركزية:

بالطبع تحقق المنظومة غاية أهدافها إذا ركزت السياسة المستخدمة على أستغلال المشغل بأعلى درجة متاحة وأبقت عليه مشغولاً طول الوقت.

### ٢ ـ إنتاجية المنظومة THROUGH PUT :

إذا كانت وحدة التشغيل المركزية تعمل فهى مشغولة وإذا كانت مشغولة فهى تنتج لذلك فإن معيار THROUGH PUT أحد المعايير الهامة ، ويعنى عدد المهام التى يمكن إنجازها خلال وحدة زمن (ساعة مثلا) وبالطبع فأن المعيار يتوقف على حجم المهام المستخدمة للمنظومة.

### ٣ ـ TURN AROUND TIME (T.A.T) :

أتخذ معيار TAT لتلافى عدم دقة المعيار السابق وتقاس T .A .T بحساب النسبة بين زمن الإنتظار منذ أستقبال المهمة وحتى تسليمها:

$$T.A.T = \frac{\text{WAIT TIME}}{\text{TOTAL TIME}}$$

### ٤ ـ زمن الإنتظار:

الواقع أن سياسة التنسيق لا تؤثر مطلقاً على زمن المعالجة أو زمن تنفيذ I/O أنما تحاول تقليل زمن الإنتظار لكل مهمة على حدة.

٥ ــ زمن الإستجابة RESPONSE TIME (R.T) :

فى الانظمة النشطة لا يعتد بمعيار T .A .T لذلك تقاس كفاءة سياســـة دوارة روبين R.R بحساب زمن الإستجابة أى قياس الوقت المستهلك فيما بين طلب المعلومة وأول ظهور بيانات الإستجابة على الشاشة.

ونخلص من هذا إلى عدة نقاط :

أ ــ تعظيم إتاحة المشغل وإنتاجية المنظومة.

ب ــ تقليل T .A .T.

ج ــ تقليل زمن الإنتظار.

د ــ تقليل زمن الإستجابة لأدنى قدر ممكن.

ويفضل فى النظم النشطة ليس تقليل متوسط زمن الاستجابة بها لكن تقليص التغاير VARIANCE (راجع مفهوم التغاير فى كتب الإحصاء) فى زمــن الإســتجابة بديلاً عن تقليص متوسط MEAN زمن الإستجابة مما يعنى تقديم خدمة أفضل لكافــة عملاء المنظومة.

## برامج تنسيق المهام:

الهدف من تنسيق العمل على المشغل ، تحديد المهمة الواجب تنفيذهــا فــى فترة زمنية بالأسلوب الذى يحقق أهداف المنظومة ويرتقى بأدائها ويزيد معوليتــها ، وإذا كان الإنسان ينسق مهامه كما أسلفنا فى مقدمة الباب ــ فإن نظم التشغيل تتولى تنسيق مهامها بأستخدام ثلاث برامج تنسيقية وفق معيار أستخدامها:

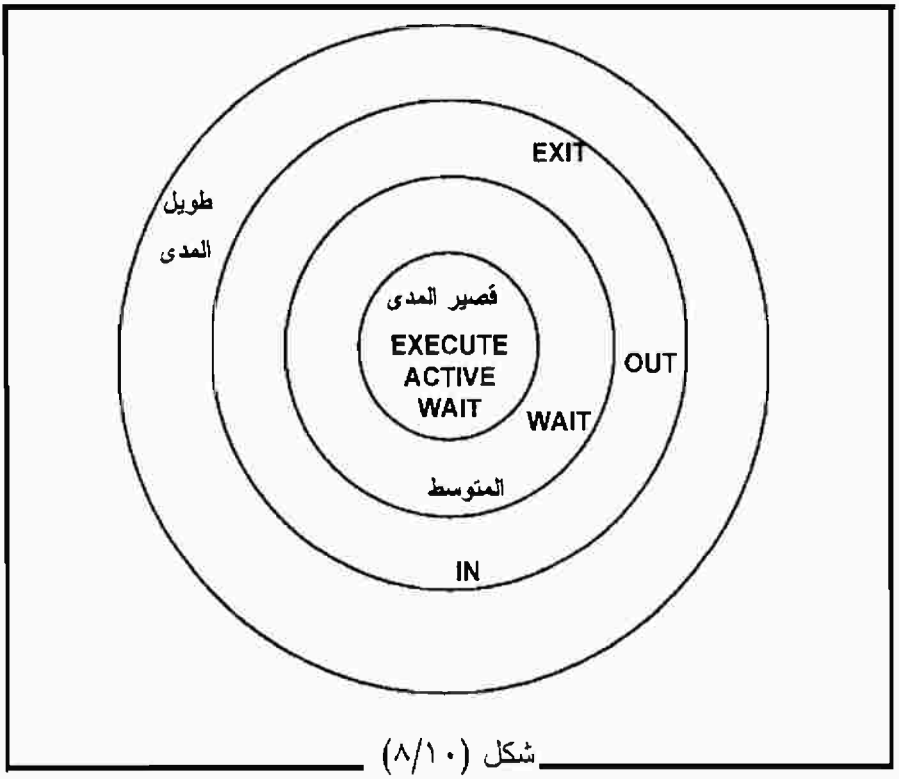
أ ــ منسق طويل المدى.

ب ــ منسق متوسط المدى.

ج ــ منسق قصير المدى.

وغاية وهدف المنسق طويل المدى أعداد البرنامج ليتحول إلى مهمة بتحديد كل المتطلبات من الكيانين المادى والبرمجى وأعداد PCB وإدراجــــها فــى قائمــة

الإنتظار النشطة ، أما المنسق قصير المدى فيعتبر المسئول الأول عن إتاحة CPU للبرنامج النشط ACTIVE المحقق لشروط التنفيذ ويوضح الشكل (٨/١٠) دالات وفئات المنسقات.




شكل (٨/١٠)

**تنسيق المهمة فى أسلوب الحيز المتصل ومستخدم واحد:**

**١ - سياسة FCFS :**

ويطبق فيه سياسة من أتى أولاً يخدم أولاً وهو صياغة مناسبة لمبدأ FIFO فى أدارة الذاكرة الأساسية ، وهو مبدأ بسيط وسهل التنفيذ.

**مثال (١):**

ورد إلى مركز الحاسب ثلاث مهام وفق الجدول التالى:

أ ــ المهمة الأولى تصل إلى المركز الساعة ١٠ وتتطلب ساعتين تشغيل.

ب ــ المهمة الثانية تصل إلى المركز الساعة ١٠،١٠ وتتطلب ساعة تشغيل.

ج ــ المهمة الثالثة تصل إلى المركز الساعة ١٠،٢٥ وتحتاج ٠،٢٥ ساعة تشغيل.

| رقم المهمة | توقيت الإستلام | زمن التشغيل اللازم / ساعة |
|---|---|---|
| ١ | ١٠ | ٢ |
| ٢ | ١٠،١٠ | ١ |
| ٣ | ١٠،٢٥ | ٠،٢٥ |

بتطبيق مبدأ FCFS يتخذ توزيع المهام الجدول التالى:

| رقم المهمة | توقيت الإستلام | توقيت بداية التنفيذ | توقيت نهاية التنفيذ | الزمن المستهلك بين إستلام المهمة وتمام تنفيذها |
|---|---|---|---|---|
| ١ | ١٠ | ١٠ | ١٢ | ٢ |
| ٢ | ١٠،١٠ | ١٢ | ١٣ | ٢،٩ |
| ٣ | ١٠،٢٥ | ١٣ | ١٣،٢٥ | ٣ |

متوسط الزمن المستهلك فى تنفيذ المهمة الواحدة

= ٧،٩ ÷ ٣ = ٢،٦٣ ساعة.

**٢ – سياسة الأسبقية للوقت الأدنى SHORTEST JOB FIRST :**

وهو أسلوب يعتبر من أبسط الطرق فى أستخدام المشغل فى ظروف تشغيل الدفعة.

**مثال (٢):**

نطبق هذه السياسة على البرامج الثلاثة السابقة :

| رقم المهمة | الإستلام | التوقيتات | | الفترة الزمنية بين البداية والنهاية |
|---|---|---|---|---|
| | | بداية التنفيذ | نهاية التنفيذ | |
| ١ | ١٠ | ١٠ | ١٢ | ٢ |
| ٢ | ١٢،٢٥ | ١٢ | ١٢،٢٥ | ٢ |
| ٣ | ١٠،١٠ | ١٢،٢٥ | ١٣،٢٥ | ٣،١٥ |

أذن متوسط الزمن المستهلك = ٧،١٥ ÷ ٣ = ٢،٣٨ ساعة.

متوسط الزمن الفعلى على الحاسب = ٣،٢٥ ÷ ٣ = ١،٠٨ ساعة.

متوسط الزمن الضائع لكل مهمة على الحاسب = $\frac{٢،٣٨}{١،٣٠}$ = ساعة

**والسؤال الآن :**

هل توجد سياسة تقلل من الزمن الضائع؟

هنا يجب النظر إلى عدة نقاط :

أ ــ إمكانية تطبيق سياسات التشغيل والتى لا تختلف كثيراً عن سياسات ادارة الذاكرة الأساسية.

ب ــ خبرة مدير تشغيل المركز حول بيئة التشغيل ومتطلبات العملاء مما قد يدفعه إلى أستخدام سابق خبرته بدلاً من تطبيق السياسات المشار إليها سلفاً .

### ٣ – سياسة أستخدام المعرفة المسبقة عن مستخدمى النظام:

إستناداً إلى المثال السابق ومن خبرة المدير التنفيذى لمركز المعلومات يمكن القول أن المركز ينتظر اليوم ثلاث مهام ، المهمة الثانية والمهمة الثالثة لا يستغرق تنفيذهما على الحاسب سوى فترات محددة ، والسؤال هل يطبق المدير التنفيذى المبدأ الأول أو المبدأ الثانى حيال المهمة الأولى (رقم ١) التى تستغرق أطول وقت على الحاسب.

هنا يلجأ المدير التنفيذى إلى الآتى:

أ ــ إنتظار ورود المهام الثانية والثالثة.

ب ــ يبدأ فى تنفيذ المهمة رقم (٣) أولاً.

ج ــ ثم ينفذ المهمة رقم (٢).

د ــ بعدها ينفذ المهمة رقم (١).

ويتضح هذا فى الجدول التالى:

| رقم المهمة | رقم التنفيذ | وصول المهمة | التوقيتات | | الوقت على الحاسب |
|---|---|---|---|---|---|
| | | | بداية التنفيذ | نهاية التنفيذ | |
| ١ | ٣ | ١٠،٢٥ | ١٠،٢٥ | ١٠،٥٠ | ٠،٢٥ |
| ٢ | ٢ | ١٠،١٠ | ١٠،٥٠ | ١١،٥٠ | ١،٤٠ |
| ٣ | ١ | ١٠ , | ١١،٥٠ | ١٣،٥٠ | ٣،٥٠ |
| | | | | | ٥،١٥ ساعة |

متوسط الزمن المستهلك = ٥،١٥ ÷ ٣ = ١،٧٢ ساعة.

متوسط الزمن الفعلى على الحاسب = ٣،٢٥ ÷ ٣ = ١،٠٨ ساعة.

متوسط الزمن الضائع لكل مهمة = ١،٧٢ – ١،٠٨ = ٠،٦٤ ساعة.

## تنسيق المهام وفق تعددية البرمجة والمشاركة فى الوقت:

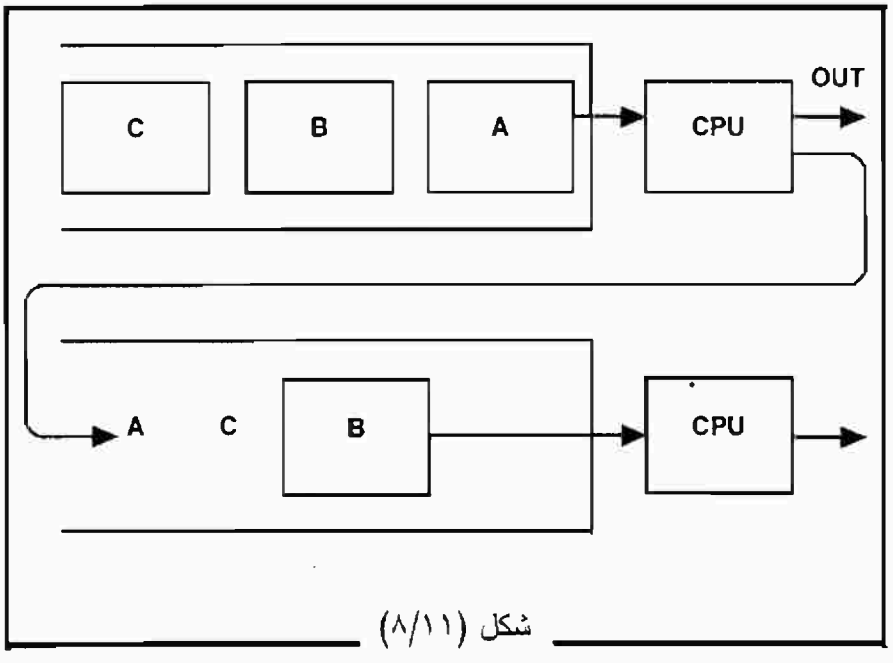
تعددية البرمجة يمكن أن نصفها بأنها نوع من المشاركة فى الوقت على مستوى وحدة التشغيل المركزية لذلك تستخدم سياسة ROUND ROBIN تعتمد على عدة مبادئ:

**الأول** : يعطى لكل برنامج شريحة من الوقت على وحدة التشغيل المركزية تسمى كوانتم فى حدود ١٠٠ ميلى ثانية بما يعادل تنفيذ ١٠٠،٠٠٠ إيعاز.

**الثانى:** يبدأ تنفيذ المهام وفق سياسة FIFO طالما ليس هناك معيار أخر.

**الثالث:** متى أستهلكت الكاونتم تتفرع وحدة التشغيل المركزية لمهمة أخرى مع وضع المهمة السابقة فى آخر قائمة الإنتظار.

ويوضح الشكل (٨/١١) مبدأ ROUND ROBIN فى أدارة وحدة التشغيل المركزية.




شكل (٨/١١)

## حيز الكوانتم الزمنى/ أو يسمى /

## حيز الكم الزمنى QUANTUM SIZE :

يعتبر تعيين المدى الزمنى للكوانتم عملية بالغة التعقيد ويدور حولها عدة تساؤلات مثل هل يفضل أن يكون الحيز كبير أم صغير؟ مع علمنا المسبق بأن تصغير المدى الزمنى يسبب مقاطعات كثيرة ويتطلب تغيير حالة وحدة التشغيل المركزية بأستمرار وبكثرة وأن تمديد المدى الزمنى يسبب تأخير تنفيذ بعض البرامج لذلك .

يمكن القول بشيء من عدم الدقة أن حيز الكوانتم فى التطبيقات التجارية للحاسبات الإلكترونية يحسب على أساس إتمام عملية I/O واحدة.

ويسجل حيز الكوانتم على نظام الساعة الداخلية للمشغل والتى تطلب إجراء مقاطعة على المشغل متى أنتهى حيز الكوانتم وبذلك تنتقل السيطرة على المشغل إلى مهمة أخرى حتى لا تستأثر مهمة بوحدة المشغل وبذلك يضمن النظام لمستخدمى تعددية البرمجة أو نظام الخدمة الفورية ON -LINE الحصول على خدمة منتظمة.

**مثال (٣):**

ورد إلى مركز الحاسب خمس مهام بياناتها كالآتى:

| رقم المهمة | التوقيتات | |
|---|---|---|
| | الوصول | التنفيذ بالساعة |
| ١ | ١٠ | ٠،٣ |
| ٢ | ٢،١٠ | ٠،٥ |
| ٣ | ١٠،٤ | ٠،١ |
| ٤ | ١٠،٥ | ٠،٤ |
| ٥ | ١٠،٨ | ٠،١ |

والمطلوب حساب ثوابت التشغيل وفق نظام الدفعـــة ومبـدأى ROUND ROBIN و FIFO.

**الحل:**

| المهمة | الإستلام | بدء التنفيذ | نهاية التنفيذ | إجمالى الوقت | (*) M : الوقت الضائع / وقت التشغيل |
|---|---|---|---|---|---|
| ١ | ١٠ | ١٠ | ١٠،٣٠ | ٠،٣ | ١ |
| ٢ | ١٠،٢ | ١٠،٣ | ١٠،٨ | ٠،٦ | ١،٢ |
| ٣ | ١٠،٤ | ١٠،٨ | ١٠،٩ | ٠،٥ | ٥ |
| ٤ | ١٠،٥ | ١٠،٩ | ١١،٣ | ٠،٨ | ٢ |
| ٥ | ١٠،٨ | ١١،٣ | ١١،٤ | ٠،٦ | ٦ |
| | | | الإجمالى | ٢،٨ | ١٥،٢ |

حساب المتوسطات:

متوسط الوقت الضائع = ٢،٨ ÷ ٥ = ٠،٥٦ ساعة.

المتوسط للمعامل M = ١٥،٢ ÷ ٥ = ٣،٠٤ ساعة.

**(*) الحقل TURN AROUND TIME (M) :**

يوضح الحقل المعنون بالوقت الضائع ، كفاءة التنسيق بين المـــهام بحيـث يستحيل أن يكون الوقت الضائع أقل من زمن التشغيل الفعلى ، ففى مهمة تســتغرق زمناً حقيقياً دقيقة فإن الوقت الضائع يكون دقيقة إذا تم تنفيذ المهمة فوراً.

لذلك أقترح المعامل M لقياس كفاءة التنسيق بالنسبة للوقت الضائع.

**فرضيات ROUND - ROBIN:**

أ ــ نفترض أن هناك مهمتين فى الذاكرة.

ب ــ وبفرض أن كل برنامج يصل بعد دقيقة من بدء تنفيذ المهمة السابقة فبفرض أن المهمة رقم (١) وصلت الساعة ١٠ وبدأت تنفيذها فوراً ثم جاءت المهمة الثانية الساعة ١٠،٢ فى حين كانت المهمة (١) مسيطرة على وحدة التشغيل المركزية فيما يوضحه الشكل (٨/١٢).

| رقم المهمة | التوقيتات | | | | |
|---|---|---|---|---|---|
| | الوقت الحقيقى | بداية التشغيل | نهاية التشغيل | الزمن الضائع | M |
| ١ | ٠،٣ | ١٠،١ | ١٠،٤ | ٠،٤ | ٣٣،١ |
| ٢ | ٠،٥ | ١٠،٢ | ١١,٣٥ | ١،١٥ | ٢،٣٠ |
| ٣ | ٠،١ | ١٠،٤ | ١٠،٦٥ | ٠،٢٥ | ٢،٥٠ |
| ٤ | ٠،٤ | ١٠،٦ | ١١،٤ | ٠،٩ | ٢،٢٥ |
| ٥ | ٠،١ | ١٠،٨ | ١١،١ | ٠،٣ | ٣ |

ونلحظ من الجدول مايلى:

أ ــ تعددية البرمجة فى بعض البرامج تسبب زيادة فى الوقت الضائع أى تسبب نقصا فى كفاءة أداء الحاسب.

ب ــ فى حين نجد أن M = ١١،٣٨ ÷ ٥ = ٢،٢٧٦ ساعة.

الزمن الضائع = ٣،٠ ÷ ٥ = ٠،٦ ساعة.

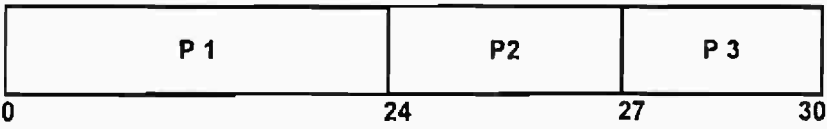
ويوضح الجدول السابق التنفيذ وفق مبدأ R - R .

**مثال (٤):**

ثلاث مهام هى P1، P 2، P 3 تستغرق على المشغل ٢٤ ، ٣ ، ٣ وحـــدة زمنية تستغرق أداء هذه المهام ، نسق تنفيذ المهام وأحسب متوسط زمن الإنتظار.

| المهمـــة | زمن التشغيل |
|---|---|
| P 1 | ٢٤ |
| P 2 | ٣ |
| P 3 | ٣ |

التنسيق وفق FIFO :

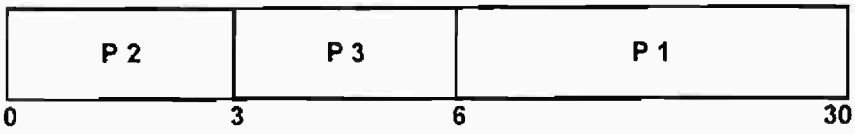



| المهمـــة | زمن الإنتظار |
|---|---|
| P 1 | صفر |
| P 2 | ٢٤ |
| P 3 | ٢٧ |

زمن الإنتظار المتوسط = $\frac{٢٤ + ٢٧}{٣}$ = ١٧

**ملحوظة:**

لو أن المهام وصلت إلى المنظومة على النحو P 2، P 3، P 1 لكـــانت معالجتها على النحو:




ومتوسط زمن الإنتظار = [ ٦ + صفر + ٣ ] ÷ ٣ = ٣ ميللى.

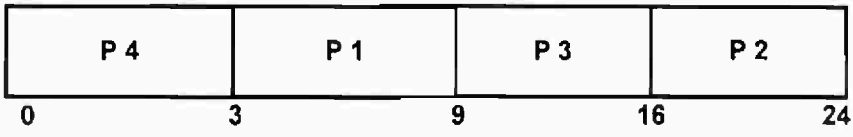
**سياسة الزمن الأدنى أولا SJF :**

**مثال (٥):**

طبق سياسة التنسيق SJF على المهام التالية ، الموضحة بالجدول:

| المهمـــة | الزمن |
|---|---|
| P 1 | ٦ |
| P 2 | ٨ |
| P 3 | ٧ |
| P 4 | ٣ |

سياسة [ SJF ] :



التأخير:

| المهمـــة | التأخير |
|---|---|
| P 4 | صفر |
| P 1 | ٣ |
| P 3 | ٩ |
| P 2 | ١٦ |
| | ٢٨ |

زمن الإنتظار المتوسط = ٢٨ / ٤ = ٧

**مثال (٦):**

فى الجدول التالى جملة مهام ترد على النحو:

| المهمـــة | زمن الوصول | زمـــن |
|---|---|---|
| P 1 | ٥ | ٨ |
| P 2 | ١ | ٤ |
| P 3 | ٢ | ٩ |
| P 4 | ٣ | ٥ |

نسق وفق [ SJF ]

فى حالتى :

PREEMPTED
NON PREEMPTED

**الحل:**

١ ــ ترد المهمة P 1 ولم تسبقها مهمة أخرى فى قائمة الإنتظار ويبدأ تنفيذها فوراً.

٢ ــ بعد وحدة زمنية واحدة تصل P٢ ونظراً لأن وقتها على CPU أقل من الباقى للمهمة P1 (٨-١) = ٧ فإنها تخلى وتدخل P2.

٣ ــ بعد دقيقة تصل P3 لكن وقتها أكبر من P2 فيبقى كما هى.

٤ ــ بعد دقيقة أخرى تصل P4 ووقتها أطول من P2.

| P 1 | P 2 | P 4 | P 1 | P 3 |
|---|---|---|---|---|

فى حالة NON PREEMPTIVE :

| P 2 | P 4 | P 1 | P |
|---|---|---|---|

**التنسيق وفق الاسبقية:**

**تمرين (٧):**

وردت المهام المبينة فى الجدول فى توقيت [صفر] نسق المهام وفق سياسة [SJF] والأسبقية:

| المهمــة | زمن الوصول | زمــن |
|---|---|---|
| P 1 | ٣ | ١٠ |
| P 2 | ١ | ١ |
| P 3 | ٣ | ٢ |
| P 4 | ٤ | ١ |
| P 5 | ٢ | ٥ |

مثال (٨) :

نسق المهام الموضحة بالجدول التالى:

| المهمة | زمن الوصول | زمن تشغيلها على CPU |
|---|---|---|
| P 1 | صفر | ٣ |
| P 2 | ٢ | ٦ |
| P 3 | ٤ | ٤ |
| P 4 | ٦ | ٥ |
| P 5 | ٨ | ٢ |

وفق مبدأ FIFO :

| زمن النهاية | ٣ | ٩ | ١٣ | ١٨ | ٢٠ | المتوسط |
|---|---|---|---|---|---|---|
| T.A.T | ٣ | ٩ | ٧ | ١٢ | ١٢ | ٨،٦ |
| TA / TS | ١ | ١،١٧ | ٢،٢٥ | ٢،٤ | ٦ | ٢،٥٦ |

وفق مبدأ SJF :

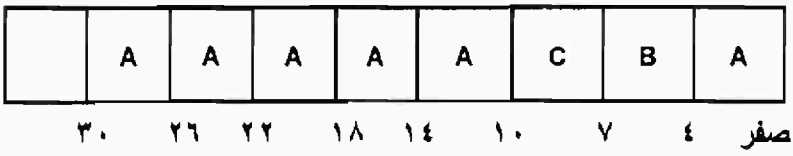
| زمن النهاية | ٣ | ١٥ | ٨ | ٢٠ | ١٠ | المتوسط |
|---|---|---|---|---|---|---|
| T.A.T | ٣ | ١٣ | ٤ | ١٤ | ٢ | ٧،٢ |
| TA / TS | ١ | ٢،١٧ | ١,- | ٢،٨ | ١ | ١،٥٩ |

**مثال (٩):**

طبق سياسة R .R على المهام التالية:

| المهمة رقم | A | B | C |
|---|---|---|---|
| زمن | ٢٤ | ٣ | ٣ |

**الحل:**




**مثال (١٠):**

نسق المهام التالية:

| P 1 | P 2 | P 3 | P 4 | P 5 |
|---|---|---|---|---|
| ١٠ | ٢٩ | ٣ | ٧ | ١٢ |

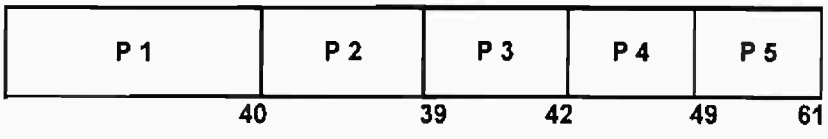
وفق FIFO

STF

R .R ( 10 UNIT , SLICE TIME)

**الحل:**

**أولا :FIFO :**



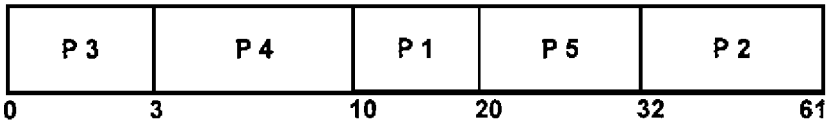
## زمن التأخير:

| | |
|---|---|
| P 1 | صفر |
| P2 | ١٠ |
| P 3 | ٣٩ |
| P 4 | ٤٩ |
| P 5 | ٤٩ |
| إجمالى | ١٤٠ |

المتوسط = ١٤٠ / ٥ = ٢٨

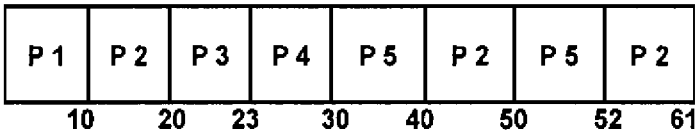
**ثانيا:STF :**

بدون إعادة إدخال.




| | |
|---|---|
| زمن التأخير | ١٠ |
| | ٣٢ |
| | صفر |
| | ٣ |
| | ٢٠ |
| المتوسط = ١٣ ميللى | 65 |

ثالثا: R.R :




أحسب زمن الإنتظار .

*