# الوحدة الثانية: إدارة الذاكرة

م. غنام الجعبري

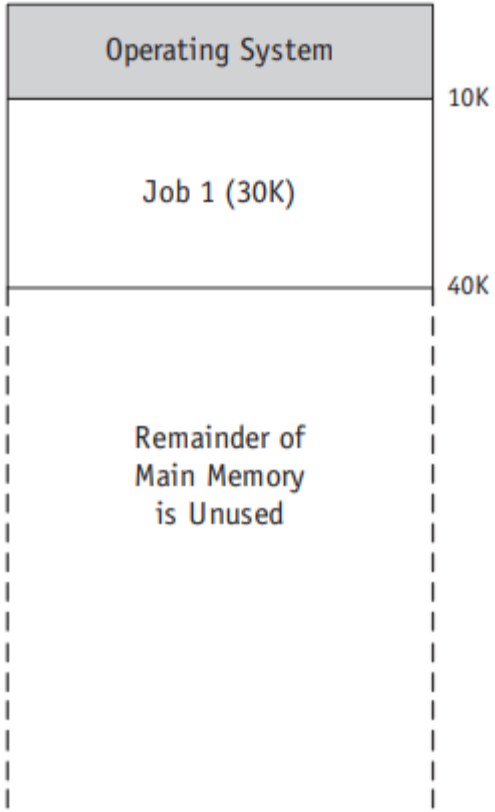
# مقدمة

- إدارة الذاكرة الرئيسية من الوظائف الهامة في نظم التشغيل لان أداء النظام ككل يعتمد على:
  - حجم الذاكرة المتوفرة
  - كيفية استغلال هذه الذاكرة بشكل افضل اثناء المعالجة
- طرق تخصيص او اشغال الذاكرة (memory allocation)
  - المستخدم الوحيد (Single-User)
  - التقسيم الثابت (Fixed Partitions)
  - التقسيم الديناميكي (Dynamic Partitions)
  - التقسيم الديناميكي القابل للنقل (Relocatable Dynamic Partitions)

# المستخدم الوحيد (Single-User)

- تدعم هذه الطريقة تخصيص او اشغال برنامج واحد في الذاكرة الرئيسية في نفس الوقت
- قبل تنفيذ البرنامج، يتم تخصيص اكبر مساحة ممكنة ومتصلة (contiguous) على الذاكرة
- اذا كان البرنامج يتطلب مساحة اكبر من المساحة المتاحة، لا يتم تنفيذ البرنامج
- لتخصيص مساحة على الذاكرة باستخدام هذه الطريقة:
  - يتم فحص المهمة الواردة اذا كانت المساحة المتاحة على الذاكرة تتسع لها، في حالة النفي يتم رفض معالجة المهمة وفحص المهمة التي تليها
  - يتم مراقبة المساحة المشغولة على الذاكرة، عند الانتهاء من معالجة المهمة وعدم الحاجة لإشغال مساحة لها في الذاكرة يتم اتاحة الذاكرة بالكامل والانتقال الى الخطوة السابقة أي فحص المهمة التالية
- احد عيوب هذه الطريقة انها لا تدعم **البرمجة المتعددة (multiprogramming)** اي اشغال الذاكرة بعدة برامج او مهام في نفس الوقت

# المستخدم الوحيد (Single-User)



# التقسيم الثابت (Fixed Partitions)

- اول طريقة تدعم البرمجة المتعددة (multiprogramming)
- يتم تقسيم الذاكرة الرئيسية الى عدة اقسام وتعيين مهمة واحدة لكل قسم، وحجم كل قسم ثابت
- إعادة تقسيم الذاكرة يتطلب إيقاف تشغيل النظام للقيام بذلك يدويا (static)
- بمجرد تعيين قسم من الذاكرة لمهمة ما، يتم منع المهام الأخرى في الأقسام الأخرى من تجاوز حدود هذا القسم
- يتطلب تخصيص مساحة على الذاكرة باستخدام هذه الطريقة اضافة خطوة أخرى الى خوارزمية المستخدم الوحيد هي مقارنة حجم المهمة مع احجام الأقسام المتاحة او الخالية للتأكد من ملائمتها بالكامل لحجم القسم

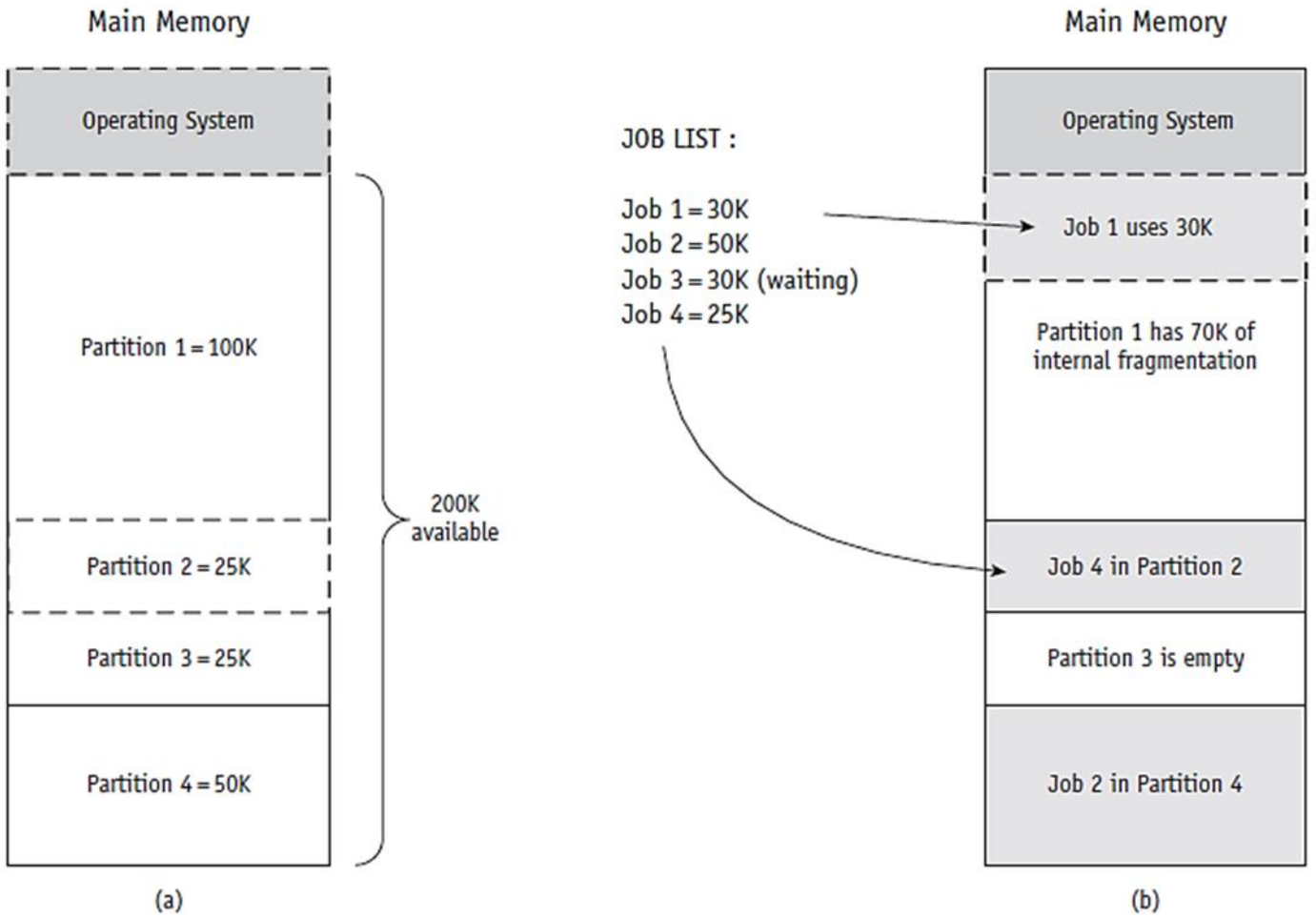
# التقسيم الثابت (Fixed Partitions)

- يجب أن يحتفظ مدير الذاكرة بجدول يوضح حجم كل قسم من أقسام الذاكرة وعنوان الذاكرة والمهمة في القسم وحالة القسم (متاح أو مشغول)

| Partition Size | Memory Address | Access | Partition Status |
|---|---|---|---|
| 100K | 200K | Job 1 | Busy |
| 25K | 300K | Job 4 | Busy |
| 25K | 325K | | Free |
| 50K | 350K | Job 2 | Busy |

- طريقة تخصيص الذاكرة لكل مهمة تعتمد على قاعدة «اول قسم متاح بالحجم المطلوب»

# التقسيم الثابت (Fixed Partitions)



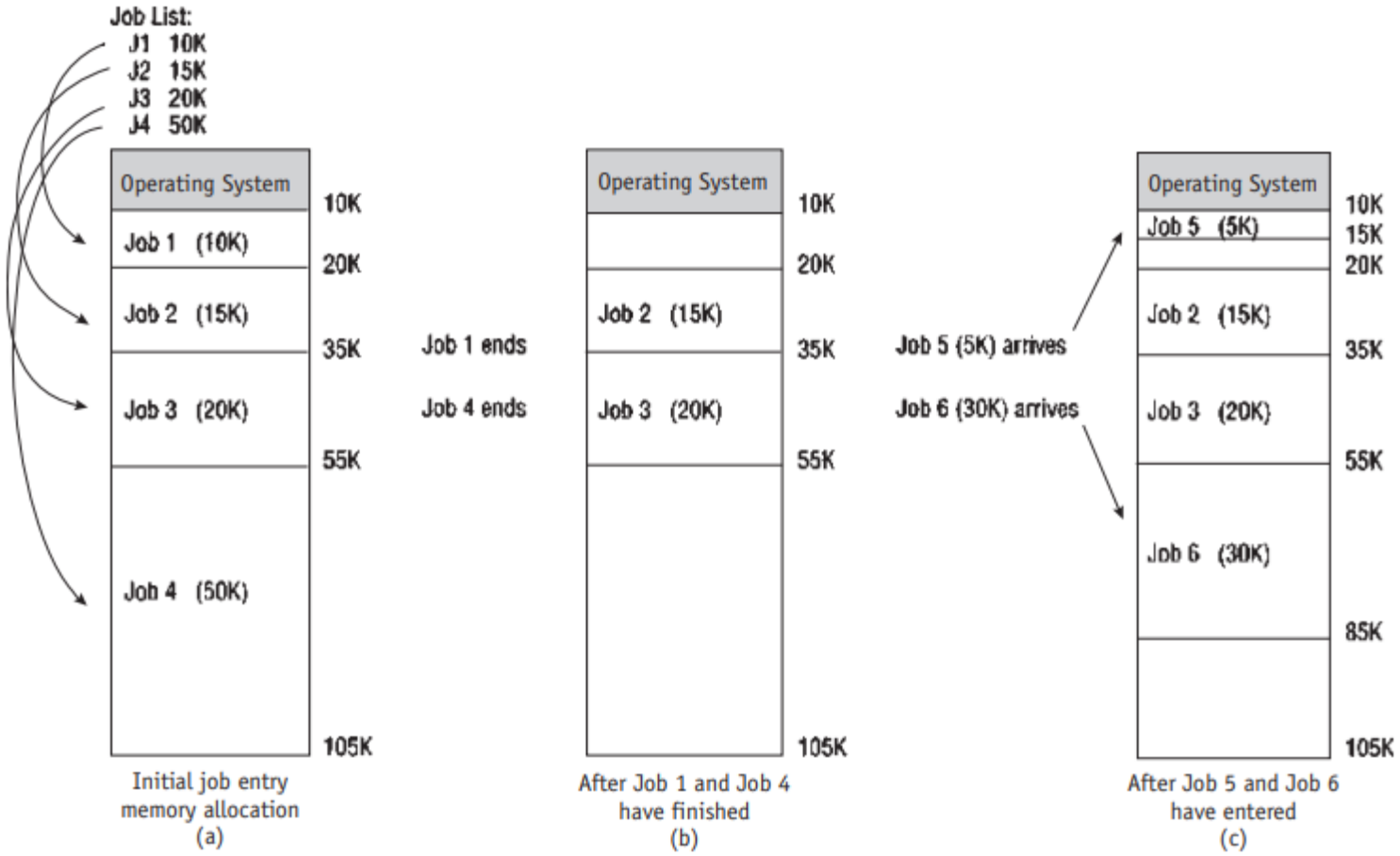
# التقسيم الثابت (Fixed Partitions)

- طريقة التقسيم الثابت تستغل الذاكرة بشكل جيد اذا كانت كافة المهام ذات احجام متماثلة او احجامها معروفة مسبقا ولا تتغير بعد إعادة تقسيم الذاكرة، لكن في معظم الاحيان يتم اختيار احجام عشوائية للأقسام بشكل لا يتناسب مع المهام الواردة
- اذا كانت الأقسام احجامها صغيرة تبقى المهام ذات الاحجام الكبيرة في حالة انتظار (waiting)
- اذا كانت كافة الأقسام لا تتسع للمهام ذات الاحجام الكبيرة يتم رفض معالجة هذه المهام
- اذا كانت الأقسام احجامها كبيرة والمهام لا تشغلها بالكامل، تنشأ مشكلة تؤدي الى ضياع المساحة على الذاكرة تدعى **التجزئة الداخلية (internal fragmentation)** لان المساحة الضائعة تكون داخل الأقسام

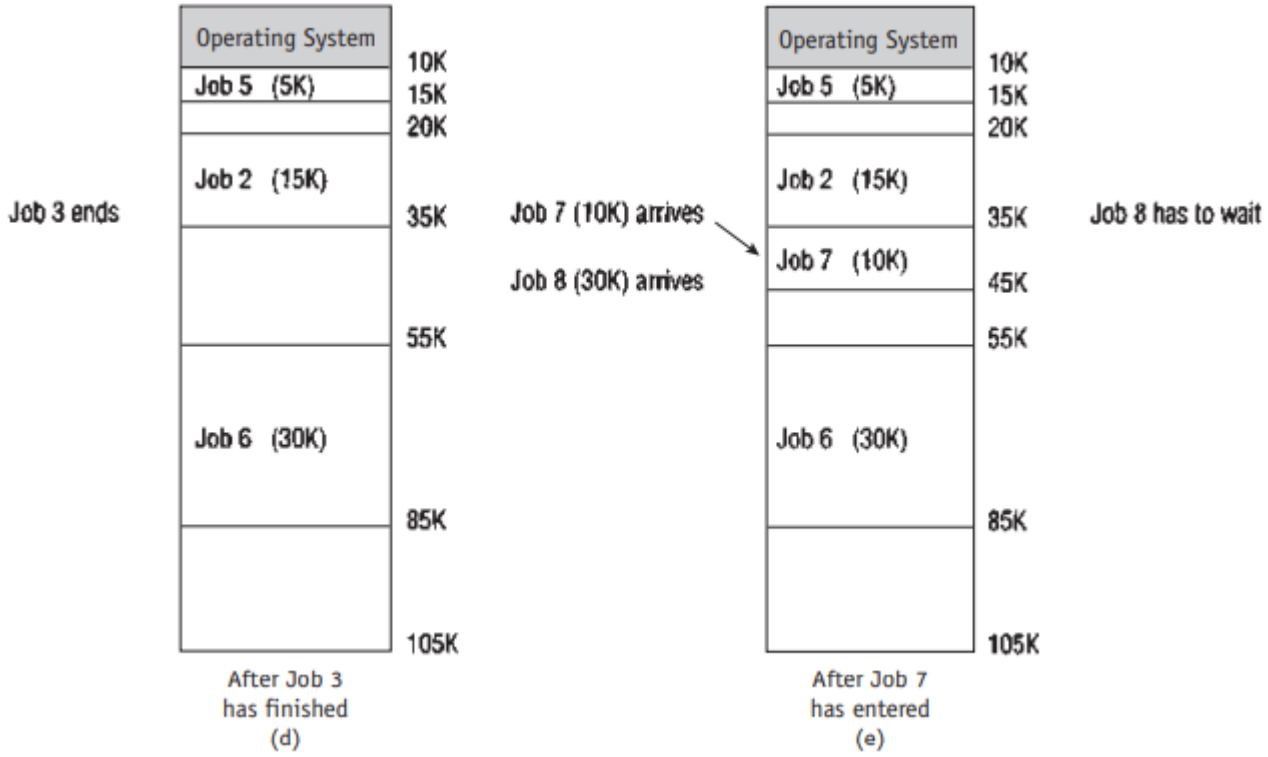
# التقسيم الديناميكي (Dynamic Partitions)

- في هذه الطريقة يتم تخصيص المساحة المطلوبة على الذاكرة لكل مهمة عند تحميلها في قسم واحد متصل (contiguous)
- طريقة تخصيص الذاكرة لكل مهمة تعتمد على قاعدة «الأولوية لمن يصل اولا» او FCFS
- حلت طريقة التقسيم الديناميكي مشكلة التجزئة الداخلية التي ظهرت في طريقة التقسيم الثابت الا انها أدت الى ظهور مشكلة أخرى
- تستغل الذاكرة بشكل كامل عند تحميل المهام لأول مرة، لكن بعد الانتهاء من معالجة هذه المهام وتحميل مهام جديدة احجامها مختلفة عن التي كانت في الذاكرة، تنشأ مشكلة تؤدي الى ضياع المساحة على الذاكرة تدعى **التجزئة الخارجية (external fragmentation)** لان المساحة الضائعة تكون بين الأقسام

# التقسيم الديناميكي (Dynamic Partitions)



# التقسيم الديناميكي (Dynamic Partitions)

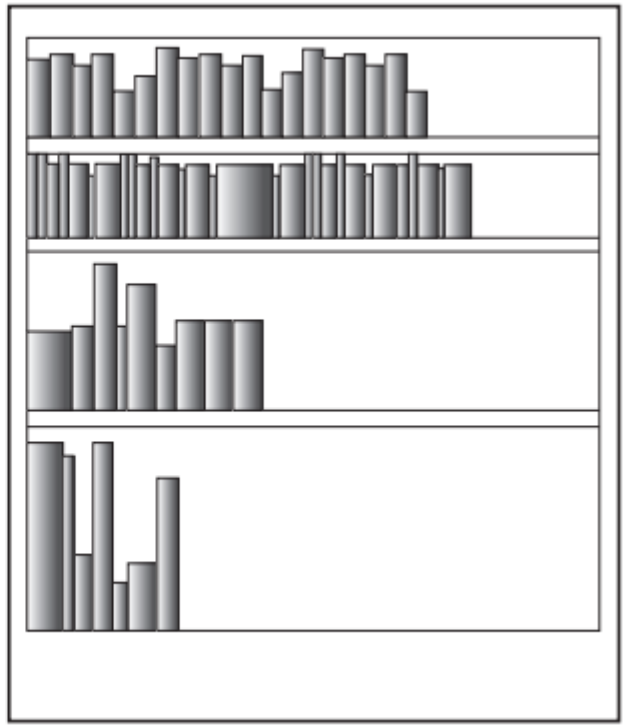


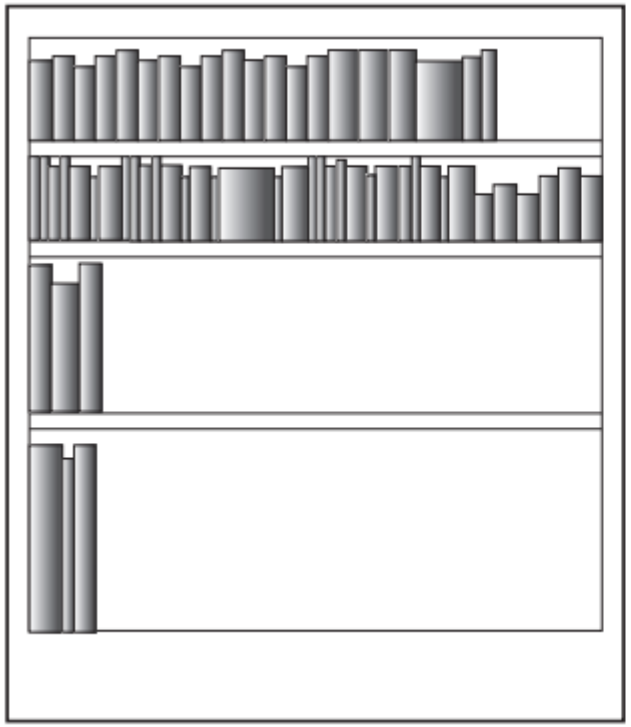
# افضل ملائمة وأول ملائمة (Best-Fit and First-Fit)

- في التقسيم الثابت والتقسيم الديناميكي، يحتفظ مدير الذاكرة بجدول عن اقسام الذاكرة وحجم كل قسم وموقعه في الذاكرة مع تحديد ان كان القسم متاحا او مشغولا
- يتم اشغال الأقسام المتاحة او الخالية اعتمادا على طريقة افضل ملائمة (best-fit) او طريقة اول ملائمة (first-fit)
- **طريقة افضل ملائمة**
  - يتم فرز الجدول حسب حجم القسم من الأصغر الى الأكبر
  - يتم تعيين المهمة في اصغر قسم بالحجم المطلوب
  - تستغل الذاكرة بشكل افضل
- **طريقة اول ملائمة**
  - يتم فرز الجدول حسب موقع الذاكرة من الادنى الى الاعلى
  - يتم تعيين المهمة في اول قسم بالحجم المطلوب
  - اسرع في الاداء

# افضل ملائمة وأول ملائمة (Best-Fit and First-Fit)

- يمكن توضيح الفرق بين الطريقتين وايهما افضل من خلال ترتيب الكتب في رفوف المكتبة
  - رفوف الكتب على اليسار مرتبة بناء على طريقة افضل ملائمة وهي افضل من حيث المساحة
  - رفوف الكتب على اليمين مرتبة بناء على طريقة اول ملائمة وهي افضل من حيث السرعة





# أول ملائمة (First-Fit)

Job List:

| Job number | Memory requested |
|---|---|
| J1 | 10K |
| J2 | 20K |
| J3 | 30K* |
| J4 | 10K |

Memory List:

| Memory location | Memory block size | Job number | Job size | Status | Internal fragmentation |
|---|---|---|---|---|---|
| 10240 | 30K | J1 | 10K | Busy | 20K |
| 40960 | 15K | J4 | 10K | Busy | 5K |
| 56320 | 50K | J2 | 20K | Busy | 30K |
| 107520 | 20K | | | Free | |
| Total Available: | 115K | Total Used: | 40K | | |

# افضل ملائمة (Best-Fit)

Job List:

| Job number | Memory requested |
|---|---|
| J1 | 10K |
| J2 | 20K |
| J3 | 30K |
| J4 | 10K |

Memory List:

| Memory location | Memory block size | Job number | Job size | Status | Internal fragmentation |
|---|---|---|---|---|---|
| 40960 | 15K | J1 | 10K | Busy | 5K |
| 107520 | 20K | J2 | 20K | Busy | None |
| 10240 | 30K | J3 | 30K | Busy | None |
| 56320 | 50K | J4 | 10K | Busy | 40K |
| Total Available: | 115K | Total Used: | 70K | | |

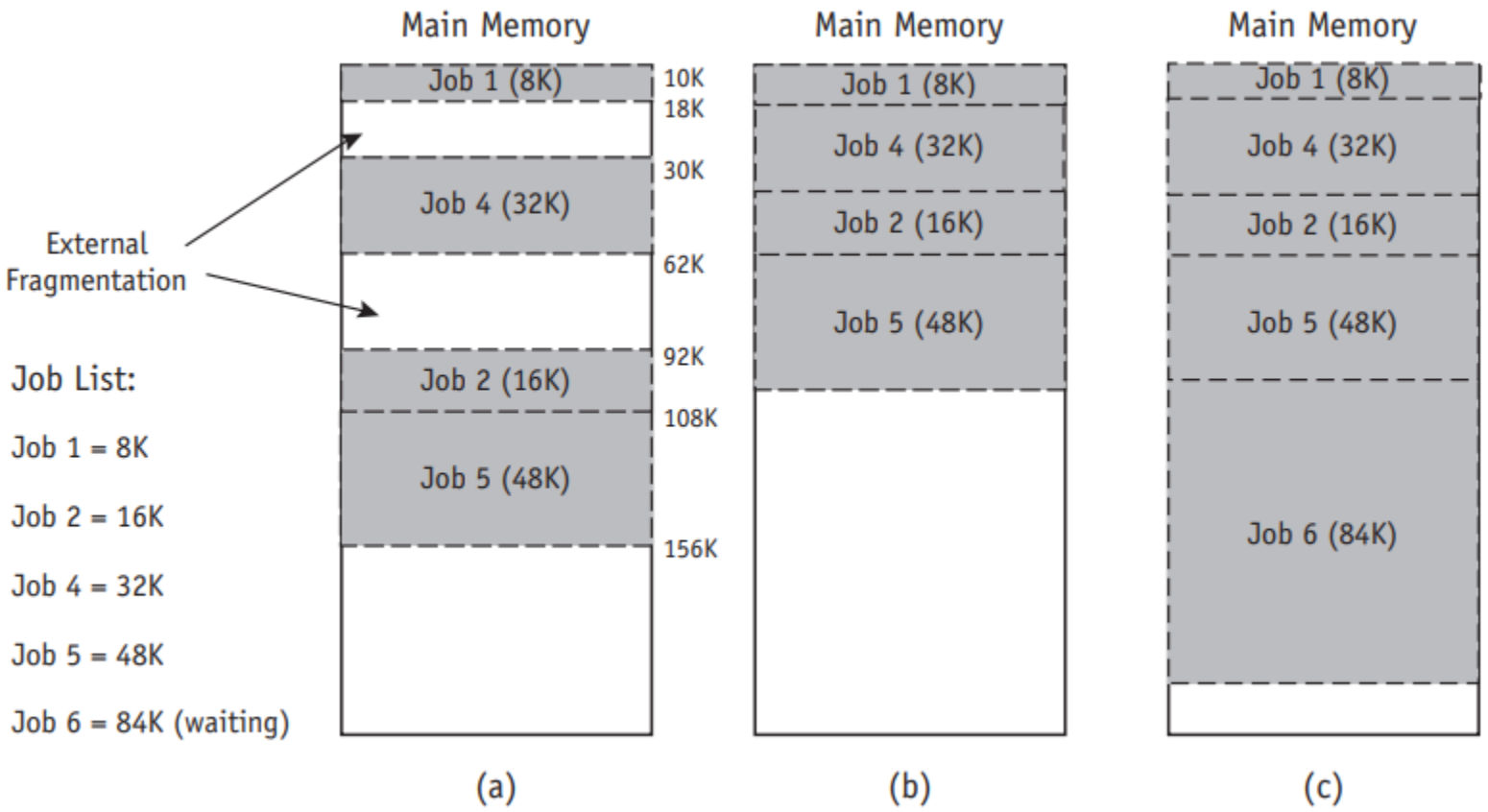
# **الغاء التخصيص** (Deallocation)

- يطلق على اتاحة مساحة من الذاكرة مصطلح **الغاء تخصيص (deallocation)**
- في التقسيم الثابت، يقوم مدير الذاكرة بإلغاء تخصيص قسم من الذاكرة او اتاحة هذا القسم بعد انتهاء المهمة عبر تغيير حالة القسم من مشغول الى متاح
- في التقسيم الديناميكي، يستخدم مدير الذاكرة خوارزمية اكثر تعقيدا في الغاء التخصيص لأنه يحاول دمج الأقسام المتاحة او الخالية مع بعضها نتيجة للحالات التالية:
  - عندما يكون القسم بعد اتاحته بجوار قسم آخر متاح
  - عندما يقع القسم بعد اتاحته بين قسمين متاحين
  - عندما يكون القسم بعد اتاحته معزولا عن الأقسام الأخرى المتاحة

# التقسيم الديناميكي القابل للنقل (Relocatable)

- في هذه الطريقة يعمل مدير الذاكرة على إعادة نقل البرامج وتجميعها مع بعض من اجل توفير اكبر مساحة ممكنة وخالية على الذاكرة لاستيعاب برامج أخرى في حالة انتظار
- وفرت حلا لمشكلة التجزئة الداخلية ومشكلة التجزئة الخارجية في الطريقتين السابقتين
- يطلق على اعادة نقل البرامج حتى تشغل مساحة متصلة على الذاكرة **إلغاء تجزئة الذاكرة (memory defragmentation)**
- الغاء تجزئة الذاكرة تعمل على إعادة تحديد موقع كل برنامج موجود في الذاكرة بحيث تكون البرامج متجاورة، وعليه يجب تعديل كل عنوان وكل إشارة الى عنوان داخل البرنامج الى عنوان الموقع الجديد في الذاكرة مع بقاء القيم الأخرى مثل بيانات البرنامج كما هي

# التقسيم الديناميكي القابل للنقل (Relocatable)



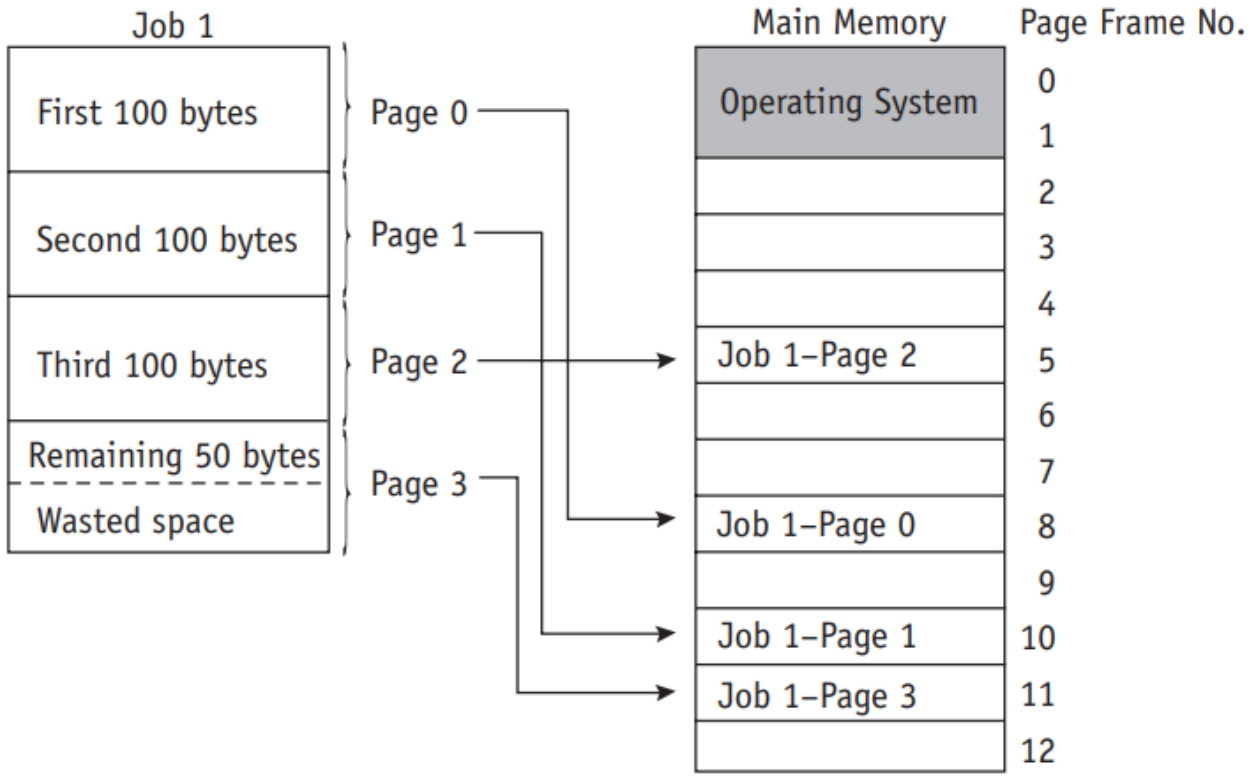
# التقسيم الديناميكي القابل للنقل (Relocatable)

- يعمل مدير الذاكرة من خلال الغاء التجزئة على تحسين استغلال الذاكرة واداء النظام، لكن السؤال: متى يقوم مدير الذاكرة بإلغاء تجزئة الذاكرة؟
- هنالك ثلاثة خيارات وكل خيار لديه حسنات وسيئات:
  - اجراء الغاء التجزئة عندما تصبح الذاكرة مشغولة بنسبة معينة مثلا 75%، لكن عيب هذا الخيار ان النظام سوف يتحمل عبء غير ضروري اذا لم تكن هنالك مهام في حالة انتظار لإشغال 25% الباقية
  - اجراء الغاء التجزئة عندما تكون هنالك مهام في حالة انتظار، وهذا يتطلب فحص دائم لقائمة المهام في حالة انتظار ويشكل عبئا اضافيا وقد يؤدي الى ابطاء معالجة المهام الحالية في النظام
  - اجراء الغاء التجزئة بعد مضي مدة زمنية محددة، لكن اذا كانت المدة قصيرة سوف يقضي النظام وقتا في الغاء التجزئة اكثر من المعالجة واذا كانت المدة طويلة كثيرا سوف يتجمع عدد كبير من المهام في القائمة وتفقد الغاء التجزئة ميزاتها

# الذاكرة الافتراضية (Virtual Memory)

- الطرق الأربعة السابقة وضعت الأساس لطرق حديثة اكثر تعقيدا في إدارة الذاكرة لأنها:
  - الغت الحاجة الى تحميل المهام على مساحة متصلة في الذاكرة، واتاحت تقسيم المهام الى عدة اجزاء وتحميلها في أماكن متفرقة على الذاكرة لكنها مرتبطة
  - الغت الحاجة الى تحميل المهام بالكامل في الذاكرة اثناء المعالجة، واتاحت تحميل جزء من المهام في الذاكرة والإبقاء على جزء آخر على القرص الصلب عند الطلب
- الطرق الحديثة اعتمدت على مفهوم **الذاكرة الافتراضية (virtual memory)** في إدارة الذاكرة

# الذاكرة الافتراضية (Virtual Memory)



# تمرين

- لديك قائمة من المهام الواردة وقائمة بالأقسام المتاحة على الذاكرة:

| المهمة (Job) | حجم الذاكرة المطلوبة |
|---|---|
| Job 1 | 690K |
| Job 2 | 275K |
| Job 3 | 760K |

| القسم (Block) | حجم القسم المتاح |
|---|---|
| Block 1 (الموقع الادنى) | 900K |
| Block 2 | 910K |
| Block 3 (الموقع الاعلى) | 300K |

- استخدم خوارزمية اول ملائمة في تخصيص او اشغال كل مهمة في الأقسام على الذاكرة
- استخدم خوارزمية افضل ملائمة في تخصيص او اشغال كل مهمة في الأقسام على الذاكرة

# تمرين

- لديك قائمة من المهام الواردة وقائمة بالأقسام المتاحة على الذاكرة:

| المهمة (Job) | حجم الذاكرة المطلوبة |
|---|---|
| Job 1 | 57K |
| Job 2 | 920K |
| Job 3 | 50K |
| Job 4 | 701K |

| القسم (Block) | حجم القسم المتاح |
|---|---|
| Block 1 (الموقع الادنى) | 900K |
| Block 2 | 910K |
| Block 3 | 200K |
| Block 4 (الموقع الاعلى) | 300K |

- استخدم خوارزمية اول ملائمة في تخصيص او اشغال كل مهمة في الأقسام على الذاكرة
- استخدم خوارزمية افضل ملائمة في تخصيص او اشغال كل مهمة في الأقسام على الذاكرة