

وظيفة عملي نظم التشغيل

محمد غانم

محمد العلوه

محمد باسل الشمالي

السكربت الأول:

خطوات الحل:

اولا يتم قراءة اسم الملف المراد نسخه الاسم يكون مع أو بدون المسار حيث تم معالجة الحالتين باستخدام التعلتين:

- `basename "$input_file_name";`
- `readlink -f "$file_name";`

بعدها نتأكد من وجود الملف الذي ادخله المستخدم:

- `if [[! -f $file_path]];then`

وفي حال لم يوجد الملف نظهر رسالة ويتوقف السكربت بتعلمية:

- `exit`

أما في حال تم العثور على الملف يُكمل السكربت، نطلب من المستخدم إدخال مسار المجلد الذي يريد وضع الملف فيه، ثم نجري اختبار لتحقق من جود ملف له نفس اسم الملف السابق باستخدام نفس التعلمية السابقة.

في حال لم يتم العثور على ملف مشابه فأنا ننفذ التعلتين:

- `cp $file_full_path $input_dir_path;`
- `chmod 775 $dir_file_path;`

في حال تم العثور على ملف مشابه ضمن المجلد المختار فأنا نعرض 4 خيارات للمستخدم وذلك باستخدام تعلمية:

- `Case`

نطلب من المستخدم ادخال رقم يعبر عن رقم الخيار للنفذه.

الخيار الأول هو استبدال الملف، في حال اختياره المستخدم نظهر رسالة تأكيد على استبدال الملف وفي حال وافق المستخدم ننفذ التعليمات التالية:

- `rm $dir_file_path;`
- `cp $file_full_path $input_dir_path;`

الخيار الثاني هو المحافظة على الملفين معاً مع تبديل الاسم، في حال تم اختيار هذا الخيار فأنا يجب أن ننشأ ملف جديد له نفس اسم الملف الموجود مع اضافة رقم الى اخره ولكننا عالجنا حالة اضافة الملف أكثر من مرة وتوليد اسماء جديدة بارقام مختلفة وكتبنا تابع يقوم بهذه العملية يمكن الاطلاع عليه من الكود، والفكرة

تقوم على البحث في المجلد عن اسماء الملفات الموجودة، وعندما ننشئ الملف الجديد نقوم بنسخ محتواه الملف الأول الى الملف الجديد وذلك باستخدام:

- cp;

الخيار الثالث هو مقارنة بين الملفين حيث تم كتابة تابع يأخذ مسار ملفين ويقوم بطباعة الاسم مع المسار والحجم والأذونات وعدد الكلمات والأسطر واستفدنا من هذا التابع في الخيار الثاني لمقارنة الملفين.

الخيار الرابع هو التراجع عن النسخ وعندها يتم اغلاق السكريبت.

السكريبت الثاني:

في البداية نطلب من المستخدم ادخال اسم ومسار المجلد، ولتحقيق المطلوب قمنا باستخدام حلقة لا نهائية تعمل كل 1 دقيقة وذلك باستخدام:

- sleep 60;

أما عن طريقة التحقق فهي تعتمد على مقارنة حجم المجلد بالبايت مع رقم معين يمثل 4 ميغا بالبايت وتم الحصول على حجم المجلد باستخدام التعليمة:

- du -b \$dir_path_input | cut -f1;

استخدمنا تعليمة قص وذلك لأن تعليمة du تعيد لنا حجم المجلد مع اسمه وبينهما تاب.

بعدها نقارن الرقم الذي انتجته التعليمة السابقة مع رقم محسوب سابقاً وفي حال كان أكبر منه:

- if [[\$file_size -gt \$max_size]]; then

فاننا نقوم بانشاء مجلد جديد له نفس اسم المجلد المختار مع اضافة archive_ وموقعه يكون بنفس المجلد الأب للمجلد الذي نقوم بمعالجته ثم نضع فيه محتويات المجلد المعالج باستخدام تعليمة:

- cp -r \$1/. \$archive_name;

ثم نقوم باستخدام التعليمة التالية للأرشفة والضغط بنفس الوقت:

- tar czf \$archive_name.tar.gz \$archive_name;

السكربت الثالث:

هذا السكربت يشابه في عمله السكربت الثاني ولكن الفرق أننا نقوم بمقارنة تاريخ تعديل الملف عوضاً عن الحجم والتعليمة التي ترد لنا تاريخ التعديل هي:

- `stat -c "%Y" $file_path;`

ولكن لاحظنا ان تعليمة تعديل الملف `cat` تقوم بتعديل التاريخ بمجرد كتابة سطر جديد، ولجعل عملية القص من الملف إلى ملف `backup` تحدث عن الانتهاء من الكتابة على الملف، اضفنا شرط إضافي هو أن لا يوجد بروسيس تعمل على الملف وحصلنا على البروسيس باستخدام:

- `ls -l $file_path;`

يمكن تجاهل المشكلة السابقة لأننا في النهاية نقص محتوى الملف ونضيفه الى ملف `backup` فلا يوجد فرق اذا قصنا سطر أو قصنا كامل الملف.

السكربت الرابع:

يبدأ السكربت بطباعة المهام النشطة وذلك باستخدام تعليمة:

- `ps -l | awk '{ printf "%s\t%s\t%s\n",$4,$8,$14 }';`

تم استخدام `awk` وذلك للعرض فقط رقم المهمة واسمها وأولويتها.

بعدها نطلب ادخال رقم المهمة ونعرض على المستخدم خمسة خيارات:

الخيار الأول هو تغيير أولوية مهمة وذلك باستخدام التعليمة:

- `renice -n $priority -p $process_id;`

الخيار الثاني إرسال إشارة `hangup` للمهمة التي اختارها وذلك باستخدام التعليمة:

- `kill -SIGHUP $process_id;`

الخيار الثالث إرسال إشارة `terminate` للمهمة وذلك باستخدام التعليمة:

- `kill -SIGTERM $process_id;`

الخيار الرابع إرسال إشارة `stop` للمهمة وذلك باستخدام التعليمة:

- `kill -SIGSTOP $process_id;`

الخيار الخامس إرسال إشارة `stop` للمهمة وذلك باستخدام التعليمة:

- `kill -SIGKILL $process_id;`