

parallel (bash)

- GNU Parallel actúa como un director de orquesta. No solo lanza los procesos, sino que los gestiona para que no colapsen el sistema:
 - Gestión de carga: Si un proceso termina, lanza el siguiente inmediatamente.
 - Orden de salida: Evita que los mensajes de error o logs de diferentes sujetos se mezclen de forma caótica en la terminal.
 - Uso de memoria: Puedes limitarlo para que no sature la RAM (vital en MRI, donde cada imagen cargada consume mucho).

```
sudo apt install parallel
```

parámetros ejecución

- `-j <núcleos>`: número de núcleos máximos a usar
- `--progress`: muestra progreso
- `--load X%`: máximo nivel de carga de la CPU antes de lanzar otro proceso
- `--memfree XG`: mínimo de memoria libre antes de lanzar otro proceso
- `--dry-run`: simulación de ejecución
- `--eta`: tiempo estimado de finalización basado en los ya ejecutados
- `--joblog`: crea un archivo de registro. Permite continuar el trabajo si se acabó inesperadamente.

parámetros argumentos

- `{}`: comodín
- `{.}`: nombre archivo sin extensión
- `{/}`: nombre de archivo sin ruta
- `{//}`: la ruta de la carpeta sin archivo
- `{/:.}`: nombre del archivo sin la ruta y sin extensión
- `:::`: lista de parámetros
- `{#}`: número de orden del proceso.

ejemplos

- compresión ficheros nii por separado:

```
parallel gzip {} ::: *.nii
```

- ```
#!/usr/bin/env bash
time ls *.nii.gz | parallel -j 8 --progress "echo Procesando {}; sleep 3"
```

- ```
#!/usr/bin/env bash

#Fitxer 1 per executar el FreeSurfer per les imatges estructurals T1

# directori d'entrada
export
SUBJECTS_INPUT_DIR=/media/analysis/usr/akaruk/ENIGMA/enigma_scz/project_81/execucio_freesurfer/input/
```

```

# on aniran els outputs, necessari mantenir nom per recon-all
export
SUBJECTS_DIR=/media/analysis/usr/akaruk/ENIGMA/enigma_scz/project_81/freesurfer_output/

# fitxer de subjectes
##export
SUBJECTS=/media/analysis/usr/akaruk/ENIGMA/enigma_scz/project_81/execucio_freesurfer/InputFreeSurfer1.txt
export
SUBJECTS=/media/analysis/usr/akaruk/ENIGMA/enigma_scz/project_53/informacio_participants/participants.txt

time cat $SUBJECTS | parallel -j 4 --progress "echo Procesando {} - {}; sleep 1"
##time cat $SUBJECTS | parallel -j 4 --progress "recon-all -i ${SUBJECTS_INPUT_DIR}{}.nii.gz -s {} -all"

exit 0
for CODE in $(cat /media/analysis/usr/akaruk/ENIGMA/enigma_scz/project_81/execucio_freesurfer/InputFreeSurfer1.txt);
do
    recon-all -i /media/analysis/usr/akaruk/ENIGMA/enigma_scz/project_81/execucio_freesurfer/input/${CODE}.nii.gz -s $CODE -all
done

```

```

#!/usr/bin/env bash

function retallaPath() {
# 1. Extraemos el nombre del archivo eliminando la ruta
    local nombre="${1##*/}"

# 2. Eliminamos la extensión .zip y devolvemos el resultado
    echo "${nombre%.zip}"
}

function unzipMove() {
    PASSWORD_ZIP="Boo5eigh"

    fitxer=$(retallaPath $1)
    echo "Procesando $fitxer";
    unzip -d ./tmp -o -P $PASSWORD_ZIP "$1"
    mv ./tmp/*.xls ./unZIPs/$fitxer.xls
}

export -f unzipMove retallaPath

time ls ./ZIPs/*.zip | parallel -j 4 --progress "unzipMove {}"

exit 0

```

```
time for f in ./ZIPs/*.zip; do
    fitxer=$(retallaPath $f)
    echo "Procesando $fitxer";
    unzip -d ./tmp -o -P $PASSWORD_ZIP "$f"
    mv ./tmp/*.xls ./unZIPs/$fitxer.xls
done
```

```
#!/usr/bin/env bash
```

```
execucio_bianca() {
```

```
    local sujeto=$1
```

```
    echo "Procesando $sujeto"
```

```
    # sleep .1 # treure!!!
```

```
    # exit    # treure!!!
```

```
    bianca \
```

```
    --singlefile=masterfile.txt \
```

```
    --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 \
```

```
    --labelfeaturenum=4 \
```

```
    --querysubjectnum=${sujeto} \
```

```
    --brainmaskfeaturenum=1 \
```

```
    --featuresubset=1,2 \
```

```
    --matfeaturenum=3 \
```

```
    --trainingpts=2000 \
```

```
    --nonlespts=10000 \
```

```
    --selectpts=noborder \
```

```
    -o resultados/s_${sujeto}_bianca_output \
```

```
    -v
```

```
    > logs/sub_${sujeto}.log 2>&1
```

```
}
```

```
mkdir -p resultados
```

```
mkdir -p logs
```

```
export -f execucio_bianca
```

```
time parallel -j 16 --progress execucio_bianca {} ::: {21..315}
```

```
# INICIO=21
```

```
# FIN=315
```

```
# time seq $INICIO $FIN | parallel -j 4 --progress "echo Procesando { }; sleep .5
```

From:

<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:

<https://miguelangel.torresegea.es/wiki/linux:bash:parallel?rev=1778239640>

Last update: **08/05/2026 04:27**

