

Laboratorio de Sistemas Operativos. Junio 2005/06. Tercero Ingeniería Informática		C1	C2	C3	C4	NOTA
Nombre:						
Apellidos:		Grupo: _____ de _____ de 20__				

1. Suponiendo que usted es usuario de una máquina UNIX, escriba los comandos para realizar las siguientes tareas:

Apartado a.- Escriba una instrucción grep que obtenga los mismos resultados que la siguiente sentencia: "grep -v '^[^a]' fichero". La sentencia grep que escriba debe tener un patrón DIFERENTE del patrón usado en la sentencia grep anterior.

```
grep '^a' fichero
```

Apartado b.- Deseamos añadir la línea "hola a todos" al principio del fichero carta.txt sin utilizar ningún editor de texto. Escriba un comando que permita realizar esto. Ojo, no debe destruir el contenido del fichero carta.txt.

Una posible solución:

```
echo "hola a todos" >> carta.txt
```

Apartado c.- Escriba un comando para obtener un listado de todas las colas de mensaje que actualmente están en uso en el sistema.

```
ipcs -q
```

Puntuación: 1'5 ptos.

Tiempo estimado: 10 min.

2. El comando *head* lee las N primeras líneas de un fichero y las envía a la salida estándar. Por ejemplo, el siguiente comando muestra por la terminal las primeras 14 líneas del archivo *prueba.c*

```
> head -n 14 prueba.c
```

El comando *tail* lee las N últimas líneas de un fichero y las envía a la salida estándar. Por ejemplo, el siguiente comando muestra por la terminal las últimas 7 líneas del archivo *prueba.c*

```
> tail -n 7 prueba.c
```

Escriba un nuevo comando llamado *selectlines* que, utilizando *head* y *tail*, muestre por pantalla el número de líneas que van desde la primera hasta la última indicada ambas inclusive. Por ejemplo, el siguiente comando muestra por la terminal desde la línea 5 hasta la línea 20 ambas inclusive. Del archivo *prueba.c*

```
> selectlines 5 20 prueba.c
```

```
/*
Llamada:
select fichero linea_orgine linea_fin
*/

#include <errno.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
```

```

int main(int argc, char *argv[]) {

    pid_t pid;
    int r;
    int l_inicio, l_fin;
    int tub[2];

    if (argc != 4) {
        printf("Uso: select fichero linea_origine linea_fin.\n");
        return -1;
    }
    l_inicio = atoi(argv[2]);
    l_fin = atoi(argv[3]);

    if (pipe(tub) == -1) {
        perror("Error creando tubería.");
        return -1;
    }

    pid = fork();
    if (pid == -1) {
        perror("Error creando primer hijo.");
        return -1;
    }
    if (pid == 0) {
        close(tub[0]);
        if (dup2(tub[1], 1) == -1) {
            perror("Error duplicando salida estándar.");
            exit(0);
        }
        close(tub[1]);

        execlp("head", "head", "-n", argv[3], argv[1], NULL);
        perror("Error ejecutando head.");
        exit(0);
    }

    pid = fork();
    if (pid == -1) {
        perror("Error creando segundo hijo");
        return -1;
    }

    if (pid == 0) {
        char tmp[255];
        close(tub[1]);
        if (dup2(tub[0], 0) == -1) {
            perror("Error duplicando entrada estándar.");
            exit(0);
        }
        close(tub[0]);

        sprintf(tmp, "%i", l_fin - l_inicio);
    }
}

```

Laboratorio de Sistemas Operativos. Junio 2005/06. Tercero Ingeniería Informática		C1	C2	C3	C4	NOTA
Nombre:		Grupo:				
Apellidos:		__ de __ de 20__				

```

// La ruta completa solo es necesaria en murillo
execlp("/usr/xpg4/bin/tail", "tail", "-n", tmp, NULL);
perror("Error ejecutando tail.");
exit(0);
}

close(tub[0]);
close(tub[1]);

pid = wait(&r);
while ( (pid != -1) ||
        ( (pid == -1) && (errno == EINTR) ) )
    pid = wait(&r);

printf("OK.\n");
return 0;
}

```

Puntuación: 2'5 ptos.
Tiempo estimado: 10 min.