

CC3301 Programación de Software de Sistemas

Tarea 4 – Semestre Primavera 2018 – Prof.: Luis Mateu

En esta tarea Ud. debe programar un servidor y un cliente para la reserva de estacionamientos por Internet. El servidor se debe lanzar por medio del comando *parking*:

\$./parking 3000

El único parámetro que recibe es el número del puerto que usa para ofrecer el servicio en el servidor. El servidor administra un estacionamiento con 5 espacios contiguos del tamaño de un auto corriente. Los espacios se enumeran como 0, 1, 2, 3 y 4.

El cliente se invoca mediante el comando *./auto*. El primer parámetro es el nombre de un chofer que desea estacionarse, el segundo corresponde a una letra que indica si está reservando un espacio (letra r) o liberándolo (letra l). En el caso de una reserva, un tercer parámetro indica cuantos espacios contiguos necesita: 1 si es un auto corriente, 2 si es una camioneta, 3 si es un camión o 5 si es un camión con acoplado. En tal caso el comando entrega el número del primer espacio otorgado. Si no están disponibles los espacios solicitados al momento de invocar *auto*, entonces este comando espera hasta que se liberen. El comando *auto* obtiene el nombre del servidor y el puerto en donde se ofrece el servicio a través de la variable de ambiente PARKING en el formato *host:puerto*, por ejemplo *anakena.dcc.uchile.cl:3000*.

La siguiente tabla muestra un ejemplo de uso de este sistema. Juan, Eva, Pato y Ana son choferes. Las filas están ordenadas cronológicamente. Se usan 5 terminales, cada uno ejecutando el shell de comandos. El directorio de trabajo inicial en cada terminal es el directorio en donde Ud. programó su tarea. Lo ingresado por el usuario aparece en **negritas**. El texto normal corresponde a lo desplegado por el programa. Además se incluyen breves frases explicativas en letra cursiva, que no aparecen realmente en la pantalla. Note que el *prompt* \$ indica exactamente cuando debe terminar un comando. Ni antes ni después.

Servidor	Juan	Eva	Pato	Ana
\$./parking 3000 01234 *1234	\$ PARKING=anakena:3000; export PARKING \$./auto juan r 1 0 \$ (A)			
***34		\$ PARKING=anakena:3000; export PARKING \$./auto eva r 2 1 \$ (B)		
			\$ PARKING=anakena:3000; export PARKING \$./auto pato r 3 (espera, C)	
****4				\$ PARKING=anakena:3000 \$ export PARKING \$./auto ana r 1 3 \$ (D)
*12*4		\$./auto eva l \$ (E)		
012*4 ****4	\$./auto juan l \$ (F)		0 \$ (G)	

En la celda (A), Juan reserva el espacio 0. En (B), Eva reserva 1 y 2. Cada vez que se reservan o liberan espacios el servidor muestra en su salida estándar los espacios que todavía están disponibles. En (C), Pato solicita 3 espacios que no hay y por lo tanto espera. En (D), Ana reserva el espacio 3. En (E), Eva libera 1 y 2. Ahora hay 3 espacios libres pero no son contiguos por lo que Pato sigue esperando. En (F), Juan libera 0. Ahora sí hay 3 espacios contiguos, 0, 1 y 2, que se otorgan a Pato y por lo tanto su espera termina (ver celda (G)).

Requerimientos

Ud. debe programar el servidor y el cliente de manera que las salidas estándares de *parking* y *auto* sean idénticas a los de la tabla de arriba y respetando su cronología. Si no se cumple este requisito se descontarán 6 puntos. No reproduzca el texto que aparece en letra cursiva y entre paréntesis. Están ahí solo para rotular las celdas.

Indicaciones

- Estudie la pregunta 1 del examen del semestre otoño 2013. Programe las funciones *reservar* y *liberar* e invoque esas funciones en el servidor.
- Use la función *getenv* para obtener el valor de una variable de ambiente. Obtenga su documentación con *man getenv*.

- Resuelva esta tarea antes del control 3. Le servirá de estudio.

Recursos

- Baje *t4.zip* de U-cursos y descomprímalo. El directorio contiene el archivo *Makefile* para compilar su tarea y los archivos *libsocket.c*, *libsocket.h*, *util.c* y *util.h* que necesitará para trabajar con sockets.
- Programe el servidor en el archivo *parking.c* y el cliente en *auto.c*.

Entrega

Ud. debe entregar un archivo .zip con *parking.c* y *auto.c* por medio de U-cursos. No incluya archivos binarios. Se descontará un punto por día de atraso. No se consideran los días sábado, domingo o festivos.